

Andrzej Bargiela

Witold Pedrycz

Granular Computing

An Introduction

**SPRINGER SCIENCE+
BUSINESS MEDIA, LLC**

GRANULAR COMPUTING

An Introduction

**THE KLUWER INTERNATIONAL SERIES
IN ENGINEERING AND COMPUTER SCIENCE**

GRANULAR COMPUTING

An Introduction

by

Andrzej Bargiela

*The Nottingham Trent University
Nottingham, United Kingdom*

Witold Pedrycz

*University of Alberta
Edmonton, AB, Canada*



SPRINGER SCIENCE+BUSINESS MEDIA, LLC

Library of Congress Cataloging-in-Publication Data

A C.I.P. Catalogue record for this book is available
from the Library of Congress.

Bargiela, Andrzej and Pedrycz, Witold

GRANULAR COMPUTING: An Introduction

ISBN 978-1-4613-5361-4 ISBN 978-1-4615-1033-8 (eBook)

DOI 10.1007/978-1-4615-1033-8

Copyright © 2003 by Springer Science+Business Media New York

Originally published by Kluwer Academic Publishers in 2003

Softcover reprint of the hardcover 1st edition 2003

All rights reserved. No part of this work may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, microfilming, recording, or otherwise, without the written permission from the Publisher, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work.

Printed on acid-free paper.

TO OUR FAMILIES

The Authors

CONTENTS

Preface

xv

PART I - METHODOLOGY AND MATHEMATICAL FRAMEWORK

Chapter 1	GRANULAR COMPUTING AS AN EMERGING PRARDIGM OF INFORMATION PROCESSING	1
1.1	Introductory comments	1
1.2	Information granules are everywhere	1
	<i>Spatial granulation: Image processing and GIS</i>	2
	<i>Temporal granulation</i>	2
1.3	Formal models of information granules	5
1.4	Conceptual aspects of information granules	6
	<i>Size of information granules and their relevance</i>	6
	<i>Usefulness of information granules</i>	7
1.5	Defining a granular world	8
1.6	Granular computing: An information processing pyramid	9
1.7	Communication between granular worlds	11
	<i>Fundamental issues of traversing information pyramid:</i>	
	<i>Encoding and decoding</i>	12
	<i>Interoperability between different formal platforms of information granules</i>	15
1.8	Conclusions	17
	References	17
Chapter 2	SETS AND INTERVALS	19
2.1	Historical background	19
2.2	The formalism of sets	22
	<i>Basic set operations</i>	23
	<i>Functional mapping of sets</i>	25
	<i>Arithmetical operations on sets</i>	27
2.3	Set enclosure	27
2.4	Interval analysis	29
	<i>Basic interval operations</i>	29
	<i>Arithmetical operations on intervals</i>	32
2.5	Interval vectors	34

2.6	Interval matrices	36
2.7	Enclosure of functions	40
	<i>Centered enclosures</i>	41
	<i>Space subdivision enclosures</i>	42
2.8	Conclusions	44
	References	45
Chapter 3	FUZZY SETS	47
3.1	The concept and formalism	47
3.2	The description and geometry of fuzzy sets	51
3.3	Main classes of membership functions	54
3.4	Operations on fuzzy sets	58
3.5	Information granularity and fuzzy sets	62
3.6	Relationships between fuzzy sets in the same space	65
3.7	Fuzzy sets and linguistic variables	66
3.8	Transformations of fuzzy sets in the same space	67
3.9	Fuzzy arithmetic	69
3.10	Fuzzy relations and relational calculus	71
3.11	Fuzzy sets and multivalued logic	74
3.12	Calibration of fuzzy sets	75
3.13	The embedding principle	76
3.14	Conclusions	77
	References	78
Chapter 4	ROUGH SETS	81
4.1	Introduction	81
4.2	The concept	81
4.3	Information systems	84
4.4	Rough sets as set approximations	87
4.5	Characterization of rough sets	88
4.6	Set comparisons in the setting of rough sets	90
4.7	Reduction of attribute spaces and reducts	92
4.8	Rough functions	93
4.9	Conclusions	95
	References	96
Chapter 5	GENERALISATIONS OF INFORMATION GRANULES	99
5.1	Interval-valued fuzzy sets	99
5.2	Fuzzy sets of type-2 and higher orders	101
5.3	Fuzzy sets of level-2 and higher	103
5.4	Fuzzy sets and rough sets	104
5.5	Shadowed sets	107
	<i>Operations on shadowed sets</i>	112
	<i>Transformations of shadowed sets</i>	113

5.6 Probabilistic sets	114
5.7 Intuitionistic fuzzy sets	115
5.8 Probability of granular constructs: Granularity and their experimental relevance	119
5.9 Concluding comments	123
References	123

PART II - ALGORITHMS OF INFORMATION GRANULATION

Chapter 6 FROM NUMBERS TO INFORMATION GRANULES	125
6.1 Introductory comments	125
6.2 Information granules and information granulation	126
6.3 The principle of granular clustering	128
<i>Conceptual design</i>	128
<i>Interpretation and validation of granular clustering</i>	130
6.4 The computational aspects of granular computing	131
<i>Defining compatibility between information granules</i>	131
<i>Expressing inclusion of information granules</i>	139
6.5 The granular analysis	141
<i>Characterization of hyperboxes</i>	142
<i>Granular feature analysis</i>	142
6.6 Experimental studies	144
<i>Synthetic data</i>	144
<i>Boston housing data</i>	151
6.7 Conclusions	158
References	159
Chapter 7 RECURSIVE INFORMATION GRANULATION	161
7.1 Introduction	161
7.2 Example application domains	162
7.3 Information granules: Design and characterization	164
<i>Building set-based information granules</i>	164
7.4 Assessment and interpretation of information granule through fuzzy clustering	174
7.5 Granular time series	179
<i>Time-domain granulation</i>	179
<i>Phase-space granulation</i>	183
7.6 Numerical studies	184
7.7 Conclusions	190
References	190

Chapter 8 GRANULAR PROTOTYPING IN FUZZY CLUSTERING	193
8.1 Introduction	193
8.2 Problem formulation	194
<i>Expressing similarity between two fuzzy sets</i>	194
<i>Performance index (objective function)</i>	196
8.3 Prototype optimisation	198
8.4 The development of granular prototypes	208
<i>Optimization of the similarity levels</i>	209
<i>An inverse similarity problem</i>	210
8.5 Conclusions	213
References	214
Chapter 9 LOGIC-BASED FUZZY CLUSTERING	217
9.1 Introduction and problem formulation	217
9.2 The algorithm	219
9.3 Experimental studies	226
9.4 Conclusions	232
References	232
Chapter 10 SEMANTICAL STABILITY OF INFORMATION GRANULES	235
10.1 Introduction	235
10.2 Information granulation: Design and validation	237
10.3 Set approximation of fuzzy sets	239
10.4 Algorithmic issues of information granulation: Design and validation	241
<i>The design of fuzzy sets - information granules</i>	241
<i>The validation phase</i>	244
10.5 Experiments	245
<i>Synthetic one-dimensional data</i>	245
<i>Real-world data</i>	248
10.6 Conclusions	253
References	253
 PART III - GRANULAR WORLD COMMUNICATIONS	
Chapter 11 COMMUNICATIONS BETWEEN GRANULAR WORLDS: FUNDAMENTALS	255
11.1 Introduction	255
11.2 Representation of fuzzy sets in the set-theoretic framework	256
11.3 Communication with a numeric world	261
11.4 Conclusions	265
References	265

Chapter 12 NETWORKING OF GRANULAR WORLDS:	
COLLABORATIVE CLUSTERING	267
12.1 Introduction	267
12.2 The horizontal collaborative clustering	270
<i>The notation</i>	270
<i>Optimization details of the collaborative clustering</i>	273
<i>The detailed clustering algorithm: A flow of computing</i>	275
<i>Quantification of the collaborative phenomenon of the clustering</i>	276
<i>Numerical examples of horizontal collaboration</i>	277
12.3 Vertical collaborative clustering	284
<i>The clustering algorithm</i>	284
<i>Numerical experiments with vertical collaboration</i>	289
12.4 Vertical and horizontal clustering: Collaboration space and data confidentiality and security	295
12.5 Conclusions	298
References	299
Chapter 13 DIRECTIONAL MODELS OF GRANULAR COMMUNICATION	301
13.1 Introduction	301
13.2 Problem formulation	302
<i>The objective function and its generalization</i>	303
<i>The logic transformation</i>	304
13.3 The algorithm	306
13.4 The overall development framework: A flow of optimisation activities	309
13.5 Experimental studies	310
13.6 Conclusions	321
References	322
Chapter 14 INTELLIGENT AGENTS AND GRANULAR WORLDS	323
14.1 Introduction	323
14.2 Communication between the agents in the granular environment	324
14.3 A fuzzy state machine as a generic model of an intelligent agent	328
14.4 The fuzzy JK flip-flop and its dynamics	330
14.5 The development of Moore type fuzzy state machines	334
<i>The architecture</i>	334
<i>A logic processor and its detailed topology</i>	335
<i>A fuzzy Moore state machine</i>	337
14.6 The learning scheme	337
14.7 Conclusions	346
References	347

PART IV - GRANULAR SYSTEMS APPLICATIONS

Chapter 15 SELF-ORGANISING MAPS IN THE DESIGN AND PROCESSING OF GRANULAR INFORMATION	349
15.1 Introduction	349
15.2 Self-organizing maps	349
<i>Revealing structure in data by cluster growing</i>	354
15.3 Associated self-organizing maps	355
<i>Weight maps</i>	355
<i>Region (clustering) map</i>	356
<i>Data distribution map</i>	357
15.4 Experiments – Synthetic and Machine Learning data	358
15.5 Case study: Analysis of software quality via software measures	364
<i>Software measures</i>	365
<i>Visualising relationships between software measures with SOMs</i>	365
15.6 Case study: A granular analysis of ECG data	369
15.7 Conclusions	375
References	376
 Chapter 16 TEMPORAL GRANULATION AND SIGNAL ANALYSIS	 377
16.1 Introductory notes	377
16.2 Granulation of signals in spatial domain	378
<i>The development of data-justifiable information granules:</i>	
<i>A formulation</i>	378
16.3 The detailed granulation algorithm	380
16.4 Granular models of signals	387
<i>Predictive description of granular models</i>	388
<i>Condensation of numeric signals</i>	388
16.5 Experimental studies	389
16.6 Rough sets in signal granulation	395
16.7 Conclusions	396
References	397
 Chapter 17 – GRANULAR DATA COMPRESSION	 399
17.1 Introduction	399
17.2 Fuzzy relational equations: A brief overview	399
17.3 Relational calculus in image compression	402
17.4 Experiments	407
17.5 Conclusions	415
References	416

Chapter 18 INTERVAL STATE ESTIMATION IN SYSTEMS MODELLING	417
18.1 Introduction	417
18.2 Estimation of the state uncertainty set	419
<i>Monte Carlo method</i>	421
<i>Linear Programming method</i>	422
<i>Ellipsoid method</i>	427
<i>Sensitivity Matrix method</i>	433
18.3 Real-life application	436
18.4 Conclusions	443
References	444
Epilogue	447
Index	449

PREFACE

This book is about Granular Computing (GC) – an emerging conceptual and computing paradigm of information processing. As the name suggests, GC concerns processing of complex information entities – information granules. In essence, information granules arise in the process of abstraction of data and derivation of knowledge from information. Information granules are everywhere. We commonly use granules of time (seconds, months, years). We granulate images; millions of pixels manipulated individually by computers appear to us as granules representing physical objects. In natural language, we operate on the basis of word-granules that become crucial entities used to realize interaction and communication between humans. Intuitively, we sense that information granules are at the heart of all our perceptual activities. In the past, several formal frameworks and tools, geared for processing specific information granules, have been proposed. Interval analysis, rough sets, fuzzy sets have all played important role in knowledge representation and processing.

Subsequently, information granulation and information granules arose in numerous application domains. Well-known ideas of rule-based systems dwell inherently on information granules. Qualitative modeling, being one of the leading threads of AI, operates on a level of information granules. Multi-tier architectures and hierarchical systems (such as those encountered in control engineering), planning and scheduling systems all exploit information granularity. We also utilize information granules when it comes to functionality granulation, reusability of information and efficient ways of developing underlying information infrastructures.

For the first time, GC brings all of these formalisms and methodologies together, treats them uniformly and becomes the fundamental computing paradigm. GC establishes a sound research agenda that promotes synergies between the already well-established technologies of sets (intervals), fuzzy sets and rough sets. Is Granular Computing a totally new pursuit? The answer is: yes and no. GC has inherited a long tradition and specialized research agenda of the contributing technologies. On the other hand, Granular Computing opens up entirely new research avenues and promotes an innovative and holistic view of the fundamental mechanisms of information representation and processing.

The past century will definitely be identified as an age of digital information processing, viewed as a synonym of the ever-present computer technology. Bits of information that are manipulated by computers are products of interval granulation. Analog-to-digital conversion is an omnipresent vehicle of communication between analog (continuous) world and the world of bits and computer hardware. What awaits us around the corner is the age of human-centered intelligent systems, WEB based virtual world, intelligent agents operating in different environments and communicating between themselves and human users. Information granules realized in the realm of Granular Computing are generic elements facilitating human-system communication and making them user-oriented. This unified and generalized platform becomes essential if we intend to address the needs of intelligent systems.

The overall material of the book is arranged into four main parts that reflect the philosophy and top-down methodology we adhere to. In Part I (Methodology and mathematical framework), we start with individual formal frameworks of information granulation such as set theory and interval analysis, fuzzy sets and rough sets. Then we move on to a series of synergistic developments that address the diversity of processes of information granulation and the information content of granules. Part II concentrates on the algorithmic facet of GC and introduces algorithms that are used to granulate information and interpret resulting granules. Clustering plays a pivotal role in this process. We propose clustering techniques that explicitly address the issue of the granular character of representatives (prototypes) discovered in experimental data. Environments of granular computing (granular worlds) exhibit an enormous variety both in terms of their level of abstraction, specificity or granulation and the formalisms of information granulation. This raises the important issue of communication between granular worlds. Part III is devoted to this matter. Finally, Part IV concentrates on the applications of GC and includes a number of case studies.

We view Granular Computing as an enabling technology and as such it cuts across a broad spectrum of disciplines and becomes important to many areas of applications. While the book provides the general theoretical framework for GC, the application part of the book includes several comprehensive case studies geared towards representative areas such as signal processing and signal representation, quality analysis in software engineering, and description of biomedical data (signals). This is intended to provide the reader with reference points for granular information processing in other applications, ranging from natural language processing through image understanding to knowledge-based system control.

The material presented in this book is self-contained. Although we expect the reader to have a basic grounding in set theory, calculus, and optimization, the familiarity required is essentially at the level of the generic notions and algorithms.

During the course of this project, a number of colleagues shared with us their views and opinions. We would like to acknowledge stimulating discussions with L. Zadeh, T.Y. Lin, V. Kreinovich, K. Hirota, H. Ishibuchi, J. Kacprzyk, I. Kosonen and M. Tanaka at the various stages of this project. Professor A. Niederlinski brought to our attention the important and innovative perspective of non-Aristotelian logic, as originally developed and promoted by Alfred Korzybski.

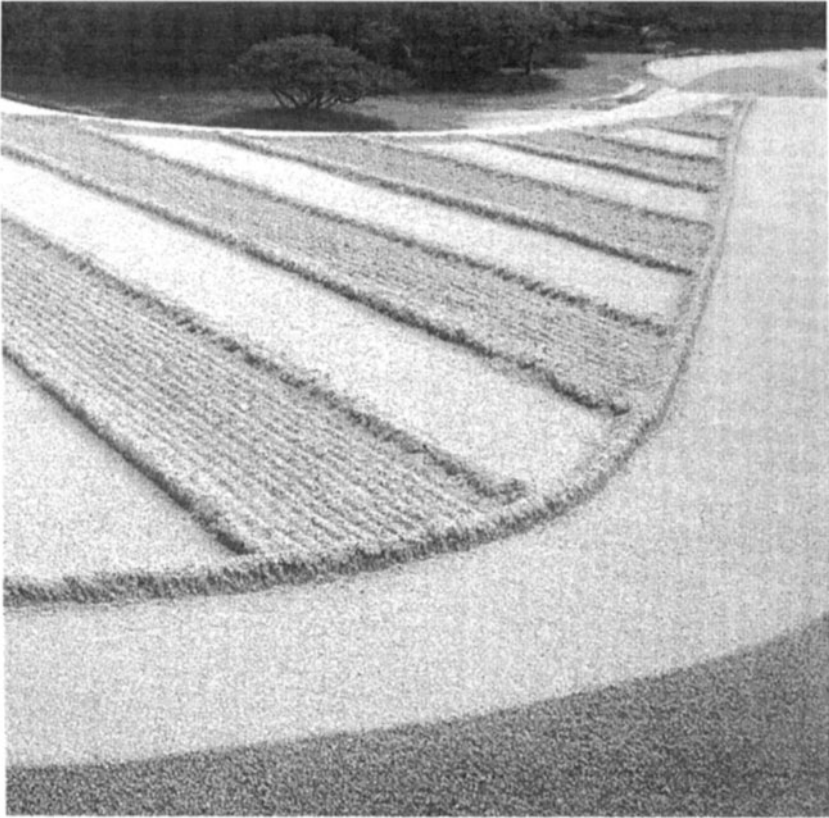
As a founder of fuzzy set theory and the originator of the concept of granular computing Professor Lotfi Zadeh provided a source of inspiration for most of the work presented in this book. His foresight about the potential of granular computing has been amply proven in many successful applications; some of which are discussed here.

We would like to acknowledge a generous support from the Engineering and Physical Sciences Research Council (EPSRC, UK), Natural Sciences and Engineering Research Council of Canada (NSERC), Canada Research Chair Program (W. Pedrycz), and Alberta Software Engineering Research Consortium (ASERC). Our words of thanks go to Susan Lagerstrom-Fife, who in her capacity of Editor provided us with all necessary professional guidance.

Andrzej Bargiela and Witold Pedrycz

PART I

METHODOLOGY AND MATHEMATICAL FRAMEWORK



GRANULAR COMPUTING AS AN EMERGING PARADIGM OF INFORMATION PROCESSING

1. 1 INTRODUCTORY COMMENTS

This book is about granular computing, its fundamentals, methodologies, algorithms and applications. In a nutshell, granular computing, as the name itself stipulates, deals with representing information in the form of some aggregates (embracing a number of individual entities) and their processing. Information granules are everywhere. They are central to processes of abstraction guiding our intellectual pursuits. Without any exaggeration one can state that processing at the level of information granules is a predominant feature of knowledge-intensive systems. This chapter serves as a concise and gentle introduction to the subject. First, we introduce the notion of information granularity through a discussion of several illustrative examples that come from commonly visible and representative areas of engineering and science. Second, we elaborate on a number of formal models of information granules and their processing. Along this line comes a discussion on the conceptual and algorithmic aspects of information granules such as their granularity, usefulness, communication and interoperability between various platforms of granular computing.

We stress that the primary intent of this chapter is to elaborate on the fundamentals and put the entire area of granular computing in a certain global perspective and prepare a stage for more detailed analysis and synthesis.

1. 2 INFORMATION GRANULES ARE EVERYWHERE

Information granules, as the name itself stipulates, are collections of entities, usually originating at the numeric level, that are arranged together due to their similarity, functional adjacency, indistinguishability, coherency or alike (Pedrycz, 2001;

Bargiela, 2001; Pedrycz and Bargiela, 2002, Zadeh, 1979, 1997; Zadeh and Kacprzyk, 1999; Pedrycz and Vukovich, 1999; Pedrycz and Smith, 1999, Pedrycz, Smith, Bargiela, 2001). Information granules as all abstraction of our reality are aimed at building efficient and user-centered views of the external world and supporting and facilitating our perception of the surrounding physical and virtual world.

Let us elaborate on some representative areas with which information granulation is inherently associated.

Spatial Granulation: Image Processing and GIS

From its very inception, image processing has been confronted with a challenging goal of building intelligent systems that are capable of understanding and describing images. This challenge is still with us and considering a rapid pace of developments at the WWW involving indexing visual information we may anticipate further quests.

Image processing naturally splits into two main and overlapping levels of processing. The lower end of the processing deals with image segmentation, edge detection, noise removal, etc. At the higher end of abstraction, we are interested in image description and interpretation. Here the level of detail (or the level of abstraction) depends on the task we have to handle. Images perceived by humans are full of information granules. An image of any landscape consists of trees, houses, roads, lakes, shrubs, etc. They are spacially distributed and this distribution is an important factor in describing the content of the image. Interestingly, all these objects are generic information granules. In many cases there are no clear boundaries between them (say forests and marshes).

Spatial granulation is central to all GIS (Geographical Information Systems) processing. Maps forms hierarchies of abstractions and granulation realizes processes of abstraction. When establishing a coarse view of the world, we deal with large information granules: continents, countries, and oceans. We are concerned with abstractions at a high level. When more details are required, we move down to regions, provinces, states, seas, etc. All minute details are revealed to us when moving down to specific maps of towns, lakes, forests, etc. The level of information granulation depends heavily on the task at hand and the need of the decision-making process.

Temporal Granulation

Time is an omnipresent variable in all human endeavors. As such its granulation is of paramount importance and happens everywhere (Dyreson et al., 2000). Granulation of time incorporate the cultural, legal, business orientation of the

designer. The granularity of time depends upon the application. On one hand, we deal with strategic planning when plans are developed based on a horizon of 10-15 years and the meaningful granules span over several years. Short term plans operate at the level of months and quarters. We talk about days when dealing with date of birth. We use very refined information granules when talking about clock cycles of a computer. In this sense, information granules carry a well-defined semantics. Collection of information granules are referred to as calendars. The hierarchy such the temporal information granules becomes evident. It becomes evident that the level of information granularity implies here an effect of indeterminacy (obviously, it occurs in any process of information granulation). For instance, when concerned about the days treated as the most detailed information granules and expressing there a birth date, it means that we encounter a “don’t know when” effect at the level of more detailed information granules (hours or minutes): we know that the person was born sometime during the given day; the precise hour or minute are not known.

Some other examples of information granules are briefly highlighted below

- in any computer system we granulate memory resources by subscribing to the notion of pages of memory as its basic operational chunks (then we may consider various swapping techniques to facilitate an efficient access to individual data items)
- In describing any problem, we tend to shy away from numbers. Instead, we tend using aggregates and building rules (*if-then statements*) that dwell on them.
- We live in an inherently analog world. Computers, by tradition and technology, perform processing in a digital world. Digitization of this nature (that dwells on set theory - interval analysis) is an example of information granulation
- All mechanisms of data compression are examples of information granulation that is carried in a certain sense

In all the examples shown above, we dealt with mechanisms of building information granules. The process of constructing information granules is referred to as information granulation. As already indicated, no matter how this granulation proceeds and what fundamental technology it involves, there are several essential factors that drive all pursuits of information granulation. Several interesting and important observations need to be made here

- A need to split the problem into a sequence of more manageable and smaller subtasks. Here granulation serves as an efficient vehicle to modularize the problem. The primary intent is to reduce an overall computing effort by exercising a fundamental principle of “*divide and conquer*” that permeates a majority of problems of system design and analysis
- A need to comprehend the problem and provide with a better insight into its essence rather than get buried in all unnecessary details. In this sense,

granulation serves as an abstraction mechanism that reduces an entire conceptual burden. As a matter of fact, by changing the “size” of the information granules, we can hide or reveal a certain amount of details one intends to deal with during a certain design phase.

- Information granulation and the ensuing processing is *human-centric* meaning that the user, designer, developer are in the center of all these endeavours. Information granules do *not* exist as tangible physical entities but they are conceptual entities that emerge in the ocean of information. Their emergence is implied by the needs of humans in a continuous quest to abstraction, summarization, condensation of information. Information granulation supports communication at different levels: between humans, humans and computers, computers and computers (the latter trend being fully reflected in terms of autonomous agents). It is needless to say that in the world of information and its various manifestations including cyberspace, processing of information granules (granular computing) becomes a necessity.

We can easily recognize that these factors occur quite ostensibly in the general categories of the problems discussed before. In all cases the abstraction and its realization in the setting of information granules becomes apparent.

On a more detailed and application – driven note, we can state that information granules may arise as a phenomenon of inherent nonuniqueness associated with the problem at hand. As a simple example, one can resort himself to any inverse problem; the type of characteristics involved (as the functions may be non-invertible) gives rise to relations and as a result, a collection of information granules rather than single numeric quantities. Dropping some input variable in a model may also lead to the same effect of granular information.

As we can observe, the concept of granular computing tends to permeate a number of significant endeavors. The reason is quite straightforward. Granular computing as opposed to numeric computing is *knowledge-oriented*. Numeric computing is *data-oriented*. Undoubtedly, knowledge-inclined processing arises as a cornerstone of data mining, rule-based models, intelligent databases, hierarchical and supervisory control, just to name a few representative examples.

The long-lasting tradition of computing using some specific information granules is a visible testimony that some specific versions of granular computing are omnipresent indeed. As a matter of fact, as we will discuss in depth, a digital – to-analog transformation leading to *digital* computing in *analog* world is just a highly representative (albeit quite specific) instance of granular computing. By tradition (and the associated technology dominant at that time), we have embarked on the digital world of computing. To interact with the continuous (analog) world, we use set-based granulation (more specifically, interval-valued granulation). This specific type of granulation comes under the name of analog-to-digital conversion.

1.3 FORMAL MODELS OF INFORMATION GRANULES

The diversity of the formal means used for information granulation and further processing of the resulting information granules has a common denominator. All of these environments share the same research agenda of the common goal to address the fundamentals of granular computing.

The process of granulation and the nature of information granules implies a certain formalism that seems to be the most suited to capture the problem at hand. Intuitively, we note a difference when building a granule of sport cars and a granule of a forest in some GIS system. In the first case the granule exhibits definite boundaries: an element (car) is either in or out. We deal with a set theory as a suitable conceptual and algorithmic framework. In the second case the situation is radically different as the term “forest” does not exhibit clearly defined boundaries: an element shown in the map may belong to the granule to some degree. Intuitively, we need a different formalism to capture the nature of such information granules.

There are a number of formal frameworks in which information granules are built. They are well-known, thoroughly investigated and coming with a vast array of applications:

- Set theory and interval analysis
- Fuzzy sets
- Rough sets
- Shadowed sets
- Probabilistic sets and probability –based granular constructs
- Higher-level granular constructs

What is worth noting, though, is a fact that most of them were developed independently without any significant interaction occurring between them.

From the general point of view, information granules defined in some space X can be treated as a mapping

$$A: X \rightarrow \mathcal{G}(X)$$

where A is an information granule of interest. \mathcal{G} denotes a formal framework of information granules. These could be sets in which case we use a notation $\mathcal{P}(X)$, fuzzy sets with the notation $\mathcal{F}(X)$, rough sets $\mathcal{K}(X)$ (Polkowski and Skowron, 1998; Lin and Cercone, 1997), shadowed sets ($\mathcal{S}(X)$) and alike. When referring to A , we always specify the framework of granulation in which A has been defined (so we say that A is an interval, fuzzy set or shadowed set).

Information granules come with underlying rules describing syntax and semantics. The semantics addresses the meaning conveyed by an information granule. As being a result of some abstraction, its meaning is well-defined and practically relevant. The syntax results directly from the formal environment in which the granules are formed. For instance, operations of aggregation such as union, intersection and others (negation, dilution, concentration, etc.) are defined as a part of the formalism being considered.

1. 4 CONCEPTUAL ASPECTS OF INFORMATION GRANULES

In spite of the diversity of the formal frameworks, information granules can be described in a fairly general form meaning that there are some general characteristics that are common across all these platforms.

Size of Information Granules and their Relevance

The question as to the definition of the “size”, “capacity” or “dimension” of the information granule is of primordial interest. How to measure granularity of the constructed information granules? How to relate this granularity with computational complexity? From the intuitive point of view, the size of the granule describes its specificity. We say how specific the granule is and how many details it embraces. The more elements we identify as belonging to the granule, the more abstract and general it becomes. Its further application implies that any model in which such information granules are used can address the problem at the corresponding level of generality (specificity). The notion of cardinality (again expressed in the pertinent language of sets, fuzzy sets, etc.) is the one commonly used. Computing the cardinality is about enumerating (counting) the number of elements in the information granule. In more detail, we may quantify granularity through an integral of the form

$$\text{Card}(A) = \int_x A(x)dx$$

where A is an information granule under consideration (being more precise, we describe A in the form pertinent to the assumed formal framework of granulation such as sets, fuzzy sets, rough sets, etc.). The higher the cardinality, the higher the abstraction of the granule and the lower its specificity. Obviously, the above expression is the simplest possible and one can think of functionals of the form

$\int_x F(A(x))dx$ where F is a certain monotonically increasing transformation of A .

Obviously in all cases we assume that such integrals do make sense.

Usefulness of Information Granules

The level of information granularity is implied by the problem in which such granules are used. We have already pointed out that information granules can be treated as conceptual building blocks with the use of which we perceive and describe the problem as well as plan some interaction with the external world (such as planning through control or decision-making or pursuing various prediction tasks). The type of description and interaction dictates the level of granularity; the most relevant (useful) level becomes selected. There could be other reasons for choosing a certain level of granularity such as e.g., a computational effort that usually is directly tied up with the size of information granules. In this sense, we regard information granules as a usefully vehicle of carry out efficient computing. All in all, one can portray this matter of usefulness of information granules versus their level of granularity is illustrated in Figure 1. We stress that the usefulness in this figure is meant in some general way as discussed above. It is noticeable that such usefulness can vary quite significantly depending upon the problem at hand: in Figure 1 (a) we witness a case where with the increasing level of granularity the decline in the usefulness level is quite limited. Figure 1 (b) alludes to the situation in which the increase in the granularity level (where we start using more detailed granules) leads to quite a substantial deterioration of the usefulness (it could well be that this is a result of excessive computing effort or too detailed information used in guidance of high-end decision-making processes).

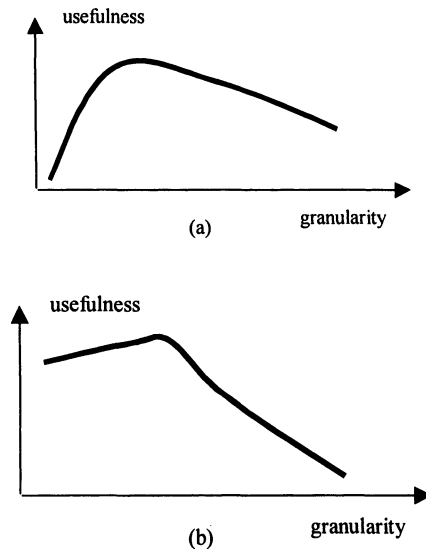


Figure 1. Usefulness of information granules as a function of their granularity: shown are two different profiles of usefulness.

1. 5 DEFINING A GRANULAR WORLD

Once we have decided upon the use of some specific formal framework, we usually define a vocabulary of granular terms that are next viewed as a frame of reference. They are just a collection of information granules with well defined semantics that we may also regard as some conceptual landmarks or so-called reference information granules. For instance, when talking about traffic on a highway, we use granules such as low, medium, high, very high speed that are further used to describe traffic patterns. Similarly, we can define a collection of some temporal information granules defined in the time domain, say morning rush, lunch time, afternoon rush, evening hours, etc. These concepts help us form a general structure of a granular world – an environment of information granules that supports all processes of information granulation, information processing and information exchange (that supports various form of interaction with the external environment). We will be building this world step by step. The family of reference information granules (a frame of reference) $\mathbf{A} = \{A_1, A_2, \dots, A_c\}$ is an important component of the granular world. So far, we have discussed the formal framework of information granules \mathcal{G} and the reference information granules (frame of reference). Now we put them together in the definition of the granular world

$$\mathbf{G} = \langle \mathbf{X}, \mathcal{G}, \mathbf{A} \dots \rangle \quad (1)$$

(the dots indicate that there are yet some more components to be defined). The syntax of operations in \mathbf{G} is completely implied by the formal framework of information granulation \mathcal{G} .

The frame of reference helps us express any information granules in the language of its elements, that is \mathbf{X} being defined in \mathbf{X} can be described in terms of A_i s. For instance, when dealing with some speed, say $X = \text{about } 80 \text{ km/hr}$, we can express this particular granule in terms of \mathbf{A} that consists of the granules such as { low speed, medium speed, high speed, very high speed}.

The frames of reference could come at different levels of granularity. For instance, one may have $\mathbf{B} = \{B_1, B_2, \dots, B_p\}$ where “p” is substantially higher than “c” granules existing in the previous frame of reference. This new frame of reference implies a new granular world, $\mathbf{G}' = \langle \mathbf{X}, \mathcal{G}, \mathbf{B} \dots \rangle$ Obviously, if we change the formal mechanism used to describe information granules, we end up with a new granular world. In the case of \mathbf{G} and \mathbf{G}' we talk about a (granular) hierarchy of the granular worlds; because of the way in which \mathbf{A} and \mathbf{B} have been formed, we say that \mathbf{G}' is a refinement of \mathbf{G} (or put it differently \mathbf{G} is an abstraction of \mathbf{G}').

Examples of relationships between granular worlds are shown in Figure 2. Note that while some of them are ordered in a linear way (because of a certain granularity of the frames of references), some others cannot be compared by having different formalism of information granulation.

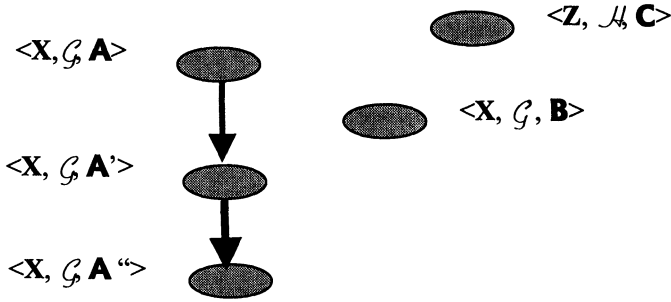


Figure 2. Relationships between granular worlds; note that some of them form a hierarchy.

1. 6 GRANULAR COMPUTING: AN INFORMATION PROCESSING PYRAMID

In granular computing we operate on information granules. As we have already noticed, information granules arise at different levels of granularity. Depending upon the problem at hand, we usually group granules of similar “size” (that is granularity) together in a single layer. If more detailed (and computationally intensive) processing is required, smaller information granules are sought. Then these granules are arranged in another layer. In total, the arrangement of this nature gives rise to the information pyramid. As portrayed schematically in Figure 3, in granular processing we encounter a number of conceptual and algorithmic layers indexed by the “size” of information granules.

Information granularity implies the usage of various techniques that are relevant for the specific level of granularity. Alluding to system modeling, we can refine Figure 4 by associating the layers of the information processing pyramid with the pertinent most commonly used classes of processing and resulting models

- at the lowest level we are concerned with numeric processing. This is a domain completely overwhelmed by numeric models such as differential equations, regression models, neural networks, etc.
- at the intermediate level we encounter larger information granules (viz. those embracing more individual elements)
- the highest level can be solely devoted to symbol-based processing and as such invokes well-known concepts of finite state machines, bond graphs, Petri nets,

qualitative simulation, etc. Note that some of these classes emerge at the intermediate level of information granularity and at that level their conceptual and symbolic fabric is usually augmented with some numeric component. It is worth stressing that the lowered granularity (higher abstraction) helps embark on models that involve logic and algebraic methods thus becoming more transparent.

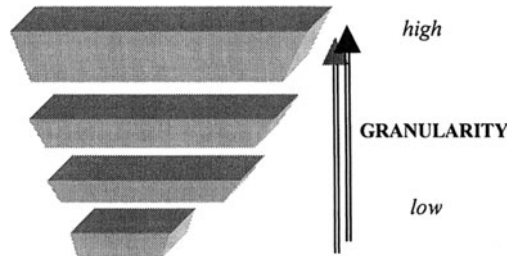


Figure 3. An information-processing pyramid (the respective layers are indexed by the corresponding level of information granularity that is granularity of the granular worlds involved).

The general characteristics of the principle of granular computing can be enumerated as shown in Table 1.

<i>Allow for multiple abstraction levels (granularity levels)</i>
<i>Allow for several methods of traversing various levels of hierarchy (encoding – decoding mechanisms)</i>
<i>Allow for nonhomogeneous methods (differential or difference equations, Petri nets, finite state machines)</i>

Table 1. The fundamental features of granular computing.

With the emergence of granular models, arise new fundamental questions as to evaluation of such constructs. Any evaluation criterion (viz. a performance index) needs to be associate with the granularity of information granules. In particular, if the model has been constructed in the setting $\langle \mathbf{X}, \mathcal{G}, \mathbf{A}' \rangle$ then its evaluation in the environment $\langle \mathbf{X}, \mathcal{G}, \mathbf{A}'' \rangle$ where \mathbf{A}'' is more specific than \mathbf{A}' may be excessively demanding. It is likely that the evaluation (and model testing) should be completed for the collection of information granules of the size that is the same and similar to those being used in the design of the model. In a similar vein, we can discuss the use of the granular model for new granular data (prediction problem) and an acceptable

level of granularity it can handle. This leads to several questions of a fundamental nature. Is the model developed with the use of “large” information granules useful when more specific results are required? It is apparent that when forming information granules, the contributing elements lose their identity that is essentially a non-recoverable process. Now, how this could effect the results of computing involving larger information granules (viz. those of lower granularity)? If we want to recover the details, how efficient could be our attempt? What are the limits of this reconstruction? These aspects boil down to the mechanisms of encoding and decoding granular information that will be discussed in the next section and becomes a part of a more general problem of communication between granular worlds.

1. 7 COMMUNICATION BETWEEN GRANULAR WORLDS

Granular worlds rarely exist and operate independently without any interaction with the environment (that could be a physical world or some other granular world). Typically, we can consider various agents each of them endowed with some granular world. The agents interact between themselves and this manifests in some form of collaboration or competition. As each agent comes with its own environment of granular computing $\langle X, \mathcal{G}, \mathbf{A} \rangle$, $\langle Y, \mathcal{G}, \mathbf{B} \rangle$, $\langle Z, \mathcal{Q}, \mathbf{C} \rangle$. To allow for any communication, one has to assure that there are some mechanisms that help agents interact with. Schematically, the communication mechanisms can be shown as a certain layer developed around the agent, Figure 4.

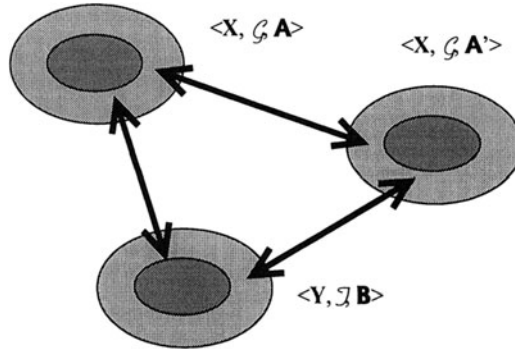


Figure 4. Collaboration between granular worlds; the mechanisms of interaction are displayed as an auxiliary processing layer around the agents.

As a consequence, the formal definition of the granular world needs to be augmented by the communication mechanisms; we add them as the family of communication procedures \mathbf{C} ,

$$\mathbf{G} = \langle X, \mathcal{G}, \mathbf{A}, \mathbf{C} \rangle \quad (2)$$

where we mean that **C** may consists of a variety of constructs that help communicate (collaborate, compete, interact with the granular worlds based on different formal schemes of information granules).

In general, we distinguish between two main categories of the communication tasks. The first one involves two granular worlds built around the same formalism of information granules that is we are concerned with $\langle X, \zeta, \mathbf{A} \rangle$ and $\langle Y, \zeta, \mathbf{B} \rangle$; the granularity of **A** and **B** can differ quite substantially. The second category of the tasks in which we do not impose any constraint on the formalism of the granular information. To elaborate on the communication mechanisms in more detail and show what they really entail, we discuss two examples: the first one arising in the context of a traversal of the information pyramid of the models we showed before and the communication between two interval-based granular environments.

Fundamental Issues of Traversing Information Pyramid: Encoding and Decoding

Granular computing supports modeling activities carried out at various levels of information granularity, refer again to Figure 5.

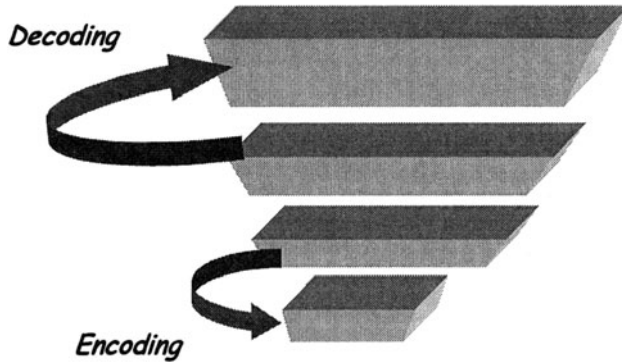


Figure 5. Decoding and encoding information granules as a vehicle of traversing the information pyramid.

The ability to traverse through the layers characterized by different sizes of information granules is one of the dominant features of the modeling pursuits discussed in this framework. Each modeling layer indexed by the assumed level of granularity, comes with its own repertoire of modeling techniques. For instance, for the highest level of information granularity, viz. numeric data, we are dealing with differential equations and regression models as basic vehicles of system modeling.

Commonly used neural networks fall under the same category. When moving towards nonnumeric layer where some information granules of lower granularity are formed, we encounter a diversity of models such as Petri nets, finite state machines, bond graphs, constraint-based, etc. (Boros et al., 2000; Harris and Brown, 1993; Kandel, 1986; Kasabov, 1996; Pedrycz, 1997; Zadeh and Kacprzyk, 1999). Depending on the specific form of granulation, we subsequently allude to fuzzy Petri nets, probabilistic Petri nets, etc.

The layers communicate between themselves. They receive data from other layers, complete computing (processing) and return the results to some other layers. These communication mechanisms are referred to as encoding and decoding, respectively. The role of the encoder is to transform the input information entering the given layer. The objective of the decoder is to convert the information granules produced by the given layer into the format acceptable by the destination layer. Depending on the problem at hand and the formalism of information granulation being used, a specific naming comes into play.

The general formulation of the encoding – decoding problem can be delineated as follows, see Figure 5: develop encoding (Enc) and associated decoding (Dec) algorithms such that the following relationship is satisfied

$$\text{Dec}(\text{Enc}(X)) = X$$

for all information granules $(X) \in \mathcal{G}(X)$ that are defined in a certain formal framework of information granulation and for a broad range of sizes of the information granules involved. In a limit case numeric granules are also included. Note, however, that the decoding-encoding scheme could be very demanding and one may not be able to meet the equality. More practically, we request that the design of these transformation should minimize the associated transformation error meaning that we are interested in minimizing the expression involving the distance $\| \cdot \|$ between the original information granule and its transformation

$$\| \text{Dec}(\text{Enc}(X)) - X \| \rightarrow \text{Min}$$

over a given range of granularity of X 's involved there and for a fixed granulation environment. As a matter of fact, the above minimization problem is not trivial. The resulting solution may very much depend upon the size of the information granules exploited in a granular world and the level of granularity of the information granule under discussion (X). intuitively, if these levels of granularity are very much distinct, the distance between X and its decoded version, $\text{Dec}(\text{Enc}(X))$ could vary quite substantially.

The A/D and D/A conversions form an interesting (yet highly specific) illustration to the formulation of the problem given above, see Figure 6. It should be stressed that

in this case the granulation process assumes a well-known version of *discretization* (let us emphasize that granulation subsumes this scheme as a particular case. Essentially, we are confined to the set-based formalism). We get:

A/D: $\text{Enc}(X) : X = \{x\} \in \mathbf{R} \rightarrow X \in \mathcal{P}(\mathbf{R})$ (the resulting granules are intervals in \mathbf{R} ; depending how the intervals are formed, one encounters either uniform quantization or a non-uniform one)

D/A: $\text{Dec}(X) : X \in \mathcal{P}(\mathbf{R}) \rightarrow X = \{x'\} \in \mathbf{R}$ (usually a quantization error occurs so we never obtain the original numeric entity, $x \neq x'$).

The A/D and D/A conversions can be revisited and generalized in the framework of fuzzy sets, $\mathcal{F}(\mathbf{R})$. This leads to the following formulation of the problem

A/D: $\text{Enc}(X) : X = \{x\} \in \mathbf{R} \rightarrow X \in \mathcal{F}(\mathbf{R})$ (the resulting granules are fuzzy sets in \mathbf{R} ; depending how they are formed, one encounters either uniform quantization or a non-uniform linguistic discretization of \mathbf{R})

D/A: $\text{Dec}(X) : X \in \mathcal{F}(\mathbf{R}) \rightarrow X = \{x'\} \in \mathbf{R}$ (usually a quantization error it can be avoided by selecting a proper family of fuzzy sets. The zero error occurs for the triangular fuzzy sets with $\frac{1}{2}$ overlap between successive membership functions).

In fuzzy controllers (Harris and Brown., 1993), the process of converting numeric data into the format accepted by the inference engine is called *fuzzification*. This is the name used for the encoding mechanism. The decoding is referred to as a *defuzzification* scheme.

One may also envision also a mixed form of information granules, namely they may originate from different formal environments of information granulation.

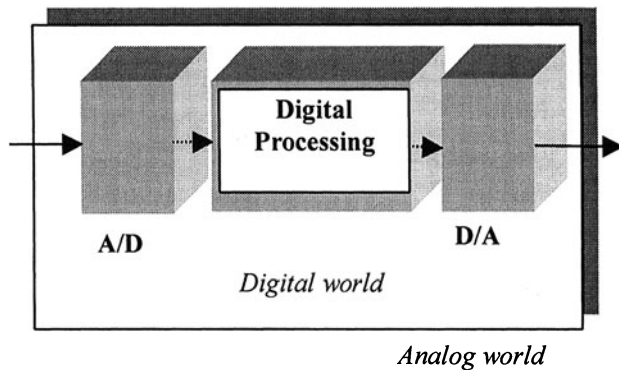


Figure 6. Digital processing as an example of commonly encountered granular computing; note a role of A/D and D/A converters utilized as the encoding and decoding modules.

Interoperability Between Different Formal Platforms of Information Granules

Various models of information granules and granulation processes themselves are crucial in the realization of interoperability when dealing with various platforms of granular computing. We illustrate this important concept in the setting of data mining or collaboration between autonomous agents. Information granules, no matter what formal framework they are supported by, are used as front and back end interfaces of the data mining computing machine. The need for the studies of the hybrid models of information granules arises when we are faced with an issue of interoperability between various tasks or subsystems of data mining that could be realized in various frameworks of granular computing. As an example, consider a situation visualized in Figure 7 (a). One data mining task, say T_1 is realized in the setting of information granules in the setting \mathcal{G}_1 . The other one is developed in the granular environment \mathcal{G}_2 . The results of the first task need to be communicated to the second module. This inherently gives rise to concept of the hybrid models of information granularity. For instance, assume that \mathcal{G}_2 dwells on set theory. Now if \mathcal{G}_1 generates the results in the form of fuzzy sets, this type of communication gives rise to fuzzy rough sets. Interestingly, even though \mathcal{G}_1 and \mathcal{G}_2 could exploit the same formalism of granular information, the communication between these two modules produces rough sets. This arises as a result of a certain level of granularity of data. As visualized in Figure 7 (b), "X is A" is a result of passing a message to the second task. This, in turn, invokes the representation of A in terms of the family of sets. As a consequence, even though we have sets at both ends, the representation of A emerges as a rough set. Put it in a different way: rough sets are just the outcome of the communication at the granular level. In more detail, refer to Figure 8, X is transformed into the following form

$$X \in P(X) \Rightarrow X^* \in R(X)$$

with **P** and **R** being the families of sets and rough sets defined in **X**. The lower and upper bound of X^* is expressed as

$$X_* = \{A_4\} \quad X^* = \{A_3, A_4, A_5\}$$

One can justify the origin and usage of some other hybrid models in the same way.

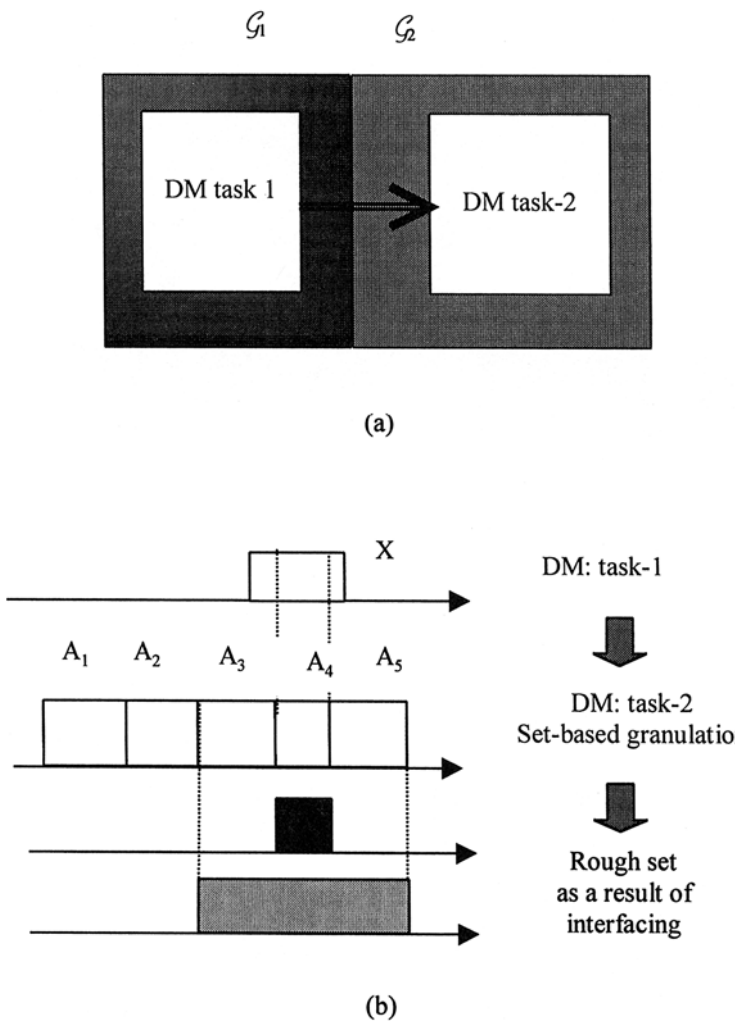


Figure 7. Two data mining tasks realized with the aid of different formalisms of information granulation: (a) a general scheme of communication, and (b) rough sets arising as an effect of communication between the two DM tasks accomplished in the granular setting implemented by sets.

1. 8 CONCLUSIONS

We have discussed the fundamentals of granular computing viewed as a new unified paradigm of processing information granules. Granular computing subsumes commonly encountered numeric processing as its special (limit) case.

The research agenda of granular computing includes a series of key and well-defined methodological and algorithmic issues

- Construction of information granules. This process deals both with the selection of the formal framework of information granulation and detailed estimation procedure producing information granules. The latter dwells on the usage of the setting in which the granules are constructed.
- Characterization of dimension (granularity) of information granules. This task is crucial as providing us with a better insight as to the essence of the granulation process and its implications both at the level of the methodology of the design of the ensuing granular model as well as its usage.
- The development of the encoding and decoding mechanisms. These are essential to the functioning of any granular architecture. The encoding and decoding schemes are essential to the performance of granular computing. Interestingly, the essence of information compatibility expressed in terms of its granularity is inherently related with granular computing and nonexistent within other environments.
- The issues of interoperability are crucial to the design of systems operating within the realm of various formalisms of information granularity.

This chapter should be treated as a bird-eye view of the rapidly growing research area. It concentrates on the methodology, attempts to identify the common features and help put the existing and somewhat scattered approaches under the same conceptual and algorithmic umbrella.

REFERENCES

- Bargiela, A. (2001) Interval and ellipsoidal uncertainty in water system state estimation, in: *Granular Computing*, (Pedrycz, W., ed.), Physica Verlag, 23-57.
- Bargiela, A., Pedrycz, W., Hirota, K. (2002), Logic-based granular prototyping, *Computers Software and Applications Conference, COMPSAC 2002*, Oxford, August 2002.
- Boros, E., Hammer, P.L., Ibaraki, T., Kogan, A., Mayoraz, E., Muchnik, I. (2000), An implementation of logical analysis of data, *IEEE Trans. on Knowledge and Data Engineering*, 12(2), 292-306.
- Dyreson, C.E., Evans, W.S., Lin, H., Snodgrass R. (2000), Efficiently supporting temporal granularities, *IEEE Trans. on Knowledge and Data Engineering*, 12(4), 568-587.
- Harris, C.J., Moore, C.G., Brown, M. (1993), *Intelligent Control - Aspects of Fuzzy Logic and Neural Nets*, World Scientific, Singapore.

Kandel, A. (1986), *Fuzzy Mathematical Techniques with Applications*, Addison-Wesley.

Kasabov, N. (1996), *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*, MIT Press, Cambridge, MA.

Lin, T.Y., Cercone N. eds.(1997), *Rough Sets and Data Mining - Analysis of Imperfect Data*, Kluwer Academic Publishers, Boston.

Pedrycz, W. (1997), *Computational Intelligence: An Introduction*, CRC Press, Boca Raton.

Pedrycz, W., Smith, M. H. (1999), Granular correlation analysis in data mining, *Proc. 18th Int Conf of the North American Fuzzy Information Processing Society (NAFIPS)*, New York, June 1-12, 715-719.

Pedrycz, W., Smith M.H., Bargiela, A. (2000), Granular clustering: A granular signature of data, *Proc. 19th Int. (IEEE) Conf. NAFIPS'2000*, Atlanta, July 2000, 69-73.

Pedrycz, W., Vukovich, G. (1999), Quantification of fuzzy mappings: a relevance of rule-based architectures, *Proc. 18th Int Conf of the North American Fuzzy Information Processing Society (NAFIPS)*, New York, June 1-12, 105-109.

Pedrycz W., ed. (2001), *Granular Computing: An Emerging Paradigm*, Physica-Verlag.

Pedrycz, W., Bargiela, A. (2002), Granular clustering: A granular signature of data, *IEEE Trans. on Systems Man and Cybernetics*, Vol 32, No. 2, 212-224.

Polkowski, L., Skowron, A. eds (1998), *Rough Sets in Knowledge Discovery*, vol.I: Methodology and Applications, Physica-Verlag, Heidelberg.

Zadeh, L. A. (1979), Fuzzy sets and information granularity, In: M.M. Gupta, R.K. Ragade, R.R. Yager, eds., *Advances in Fuzzy Set Theory and Applications*, North Holland, Amsterdam, 3-18.

Zadeh, L. A. (1997), Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, *Fuzzy Sets and Systems*, **90**, 111-117.

Zadeh, L. A., Kacprzyk, J. (1999), *Computing with Words in Information/Intelligent Systems*, vol. 1-2, Physica-Verlag, Heidelberg.

SETS AND INTERVALS

The chapter on set theory and interval analysis fulfils a dual role in this book. It provides a backdrop against which a more general representation of the world, using fuzzy sets, rough sets, shadowed sets, etc. are developed and it serves as a basis for a spectrum of granular information processing applications within which the classical set and interval formalisms are sufficiently powerful and provide valuable insights (Chapter xx). We adopt here a commonsense approach to set theory, as opposed to axiomatized approach, and focus on constructive development of granules in multi-dimensional spaces from the basic constructs of one-dimensional, connected sets, i.e. intervals.

2. 1 HISTORICAL BACKGROUND

Set theory occupies a unique place in modern mathematics since it can be shown that it can be used as a starting point for the derivation of all other branches of mathematics. This has been succinctly expressed by Bourbaki (1948):

... all mathematical theories may be regarded as extensions of the general theory of sets ... on these foundations I can state that I can build up the whole of the mathematics of the present day...

Indeed, one can represent natural numbers as a set, a rational number as a pair of natural numbers, real numbers as a set of rationals, and so on. So, the basic mathematical entities can be regarded as sets and the operations on these entities are, as a consequence, operations on sets. This brings up the possibility of using set theory as the foundations of Granular Computing or more generally, as a basis of knowledge oriented information processing. The two main reasons why this possibility is realistic is the ease with which set theory accommodates conceptual innovations and represents information of progressively increasing complexity. Another reason is that the formalism of sets seems to correlate well with our intuitions. The latter has been eloquently stated by Goedel (1940):

... despite the remoteness from sense-experience, we do have something like a perception of the objects of set theory, as is seen from the fact that the axioms force themselves upon us as being true. I don't see any reason why we should have less confidence in this kind of perception, i.e. in mathematical intuition, than sense-perception.

The origins of set theory are rather different from most other areas of mathematics. Rather than being a product of a collective intellectual effort of many people, set theory arose as a product of work of one person Georg Cantor, who published his seminal works in the period 1874-1884. The fundamental nature of set theory raised a considerable opposition from the leading mathematicians of the time but it benefited from the attention and criticism as it showed its ability to refine and adjust to cope with the paradoxes that were identified in the early versions of the theory.

One of the paradoxes discovered by Cantor himself concerned the cardinality of the “set of all sets”. On one hand this set, by definition, has the highest cardinality but the cardinality of the set of all subsets drawn from this set is bound to be higher; thus contradicting the initial assertion. Another paradox, discovered by Russel in 1902, concerns the construction of sets. Russel defined a set A as

$$A = \{X \mid X \text{ is not a member of } X\}$$

and investigated the problem whether A is an element of A . Both the assumption that A is a member of A and A is not a member of A lead to a contradiction.

The various revisions of the original set theory, that were proposed in order to overcome these paradoxes, share a common feature that they abandon a “uniform set” view of mathematical entities and introduce some form of hierarchy. This in itself is quite an intuitively pleasing development as it emphasizes the semantical transformation (qualitative change) that occurs when grouping individual elements into sets or conversely when identifying sub-sets within a given set. A good representative of the revised set theory is the one proposed by Goedel and other researchers (Goedel, 1940). In essence, the revised set theory does not consider just one type of basic constructs (i.e. sets) but is founded on three primitive notions: class, set and membership. Classes are considered as entities corresponding to some, but not necessarily all, properties. Consequently, the various classes represent conceptually different entities. If a class is a member of some other class it is considered a set, otherwise it is considered a proper class. The paradoxes that were identified earlier are avoided since the set of all sets is considered a proper class.

While the attempts to axiomatize set theory are important, we will elaborate in subsequent sections on the commonsense interpretation of the theory, which is largely build around the original notion of sets (as proposed by Cantor) augmented by the notion of classes and membership. The various axiomatic approaches are

documented in the literature, (Ackermann, 1956), (Morse, 1965), (Mendelson, 1987) and (Barwise, 1989).

Prompted by the success of set theory and the development of digital computers, the 1950s and 1960s witnessed a significant rise of interest in developing a mathematically well-defined connection between discrete computing world and the continuous real-life. Interval arithmetic provided an important generalization of arithmetic defined on real numbers. The notion of arithmetic defined on intervals (sets) first appeared in (Warmus, 1956, 1961) and (Sunaga, 1958). However, a broader research interest in interval arithmetic was stimulated by R.E. Moore's seminal dissertation (Moore, 1962). Since then, there has been a sustained research effort in developing theory, applications and the support software for interval computations (Kearfott, Kreinovich, 1996; Kreinovich et al, 1998).

The set-theoretical framework of interval computations enabled overcoming the limitations of the standard floating-point arithmetic, with its inherent round-off errors, and focused on the 'guaranteed' results of computation, i.e. computation of sets enclosing the actual solution. This proved to offer a 'natural' description of a large class of practical problems where the measurements, model parameters or computational errors can only be bound but otherwise are unknown. A general proof of existence of interval solutions to non-linear systems of equations (Hansen, 1975), (Nikel, 1981) provides a firm basis for the use of intervals in many application domains. An example of one of the most successful application of intervals to solving real-life problems is that of solving Gibbs free-energy equations arising in a super-conducting super-collider design (Stadtherr, 1994). The significance of that solution is twofold: first it showed the computational feasibility of large scale interval computations and second it provided a benchmark against which many floating-point solutions were be measured (and found to be inadequate).

The Granular Computing perspective on the use of intervals encompasses the aspect of guaranteed numerical computations but it also includes a broader concern of information abstraction. In this regard, Granular Computing capitalises on the inherent property of sets; that of being members of some classes (Goedel, 1940). The interpretability of intervals, and the higher-dimensional constructs built on intervals, provides a good basis for the derivation of abstractions of a given data set. In the subsequent sections we will provide a brief overview of set operations in general and operations on intervals in particular, introduce hyperboxes as Cartesian products of intervals in multi-dimensional spaces and will consider the quality of such an approximation measured as the discrepancy between the set enclosure and the set itself. Complementary to the development of set enclosures is the consideration of inclusion functions that can be easily evaluated while providing tight bounding of the values of real functions.

2. 2 THE FORMALISM OF SETS

Throughout this book we will denote sets by uppercase letters and the members of the sets by lowercase letters. The boldface letter \mathbf{X} will denote the *universe of discourse*, also referred to as a *universal set*. This set contains all possible elements that are relevant in each particular context. An important and frequently used universal set is the set of all points in the n -dimensional space (i.e. n -tuples of real numbers). This set is denoted as \mathbf{R}^n . The relationship of *belonging* to a set is denoted formally as $a \in A$ and the converse relationship of *not belonging* to (being excluded from) a set is denoted as $a \notin A$.

A set is defined either by listing all of its elements or by defining properties that are satisfied by all of the set elements. The first method is suitable only for specification of finite sets. For example the set A can be defined as $A = \{a_1, a_2, a_3, \dots, a_n\}$. The second method allows specification of both finite and infinite sets. Given the specific properties P_1, P_2, \dots, P_n one can define a set B as $B = \{b \mid b \text{ has properties } P_1, P_2, \dots, P_n\}$. Here the symbol " \mid " is read as "such that". Once a set has been defined it can be used as an element of a *family of sets*. This is denoted as $\{A_i \mid i \in I\}$ where I is the *set identifier* and I is the *identification set*. The number of elements that belong to set A is called the *cardinality* of the set and is denoted by $|A|$. It is clear that the cardinality of some sets, e.g. those defined through some properties, may be infinite.

Having defined sets we can move on to consider relationships between sets. One of the most basic among those is the relationship of set inclusion. If every element of set A belongs also to set B we call A a *subset* of B . This is denoted as $A \subset B$. If the converse relationship of $B \subset A$ is also true we say that sets A and B are *equal*, $A = B$. It is easy to note that every non-empty set is a subset of itself and of a universal set \mathbf{X} . The elements of \mathbf{X} that satisfy the property of *not belonging* to A form a *complement* of A and are denoted as \bar{A} . Complementation is always *involution*; that is, taking complement of a complement yields the original set. An *empty set*, i.e. a set that contains no elements, is denoted by \emptyset . The empty set is a subset of every set except itself. The complement of the empty set equals the universal set and the complement of the universal set equals the empty set.

The property of belonging, or not belonging, to a given set A can be expressed as a 2-valued function defined on all elements of the universal set \mathbf{X} . This is referred to as *characteristic function*

$$\chi_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

Although this may seem at first as an unnecessarily complex way of expressing the relationship of belonging to a set, the characteristic function provides a scope for a

generalization of the meaning of this relationship. This issue is fully discussed in Chapter 3.

A family of sets consisting of all the subsets of a particular set A is referred to as the *power set* of A and is indicated by $\mathcal{P}(A)$. The cardinality of the power set of A is always equal $2^{|A|}$.

Basic Set Operations

The *union* of sets A and B is the set containing the elements that belong to A or B or both A and B . This is denoted by $A \cup B$ and formally expressed as

$$A \cup B = \{x \mid x \in A \text{ or } x \in B\}.$$

The union of any set A with the universal set yields the universal set and the union of set A with the empty set yields the set A . Since the elements of the universal set either belong to a given set A or to its complement, the union of a set and its complement is equal to a universal set, $A \cup \bar{A} = X$. This property is usually called the *law of excluded middle*. In Chapter 2 we will elaborate on the restrictive nature of the logic that adopts this law.

The *intersection* of sets A and B is the set containing all the elements belonging to both set A and set B . This is denoted by $A \cap B$

$$A \cap B = \{x \mid x \in A \text{ and } x \in B\}$$

The intersection of any set with the universal set yields the set itself and the intersection of any set with the empty set yields the empty set. Since a set and its complement, by definition, have no elements in common, their intersection yields an empty set. This property is usually called the *law of contradiction* and is dual to the law of excluded middle.

Several important properties that are satisfied by the operations of union, intersection and complement are listed below:

Involution: $\overline{\bar{A}} = A$

Distributivity: $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
 $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

Associativity: $A \cap (B \cap C) = (A \cap B) \cap C$
 $A \cup (B \cup C) = (A \cup B) \cup C$

<i>Comutativity:</i>	$A \cup B = B \cup A$ $A \cap B = B \cap A$
<i>Absorption:</i>	$(A \cup B) \cap A = A$ $(A \cap B) \cup A = A$
<i>Idempotence:</i>	$A \cup A = A$ $A \cap A = A$
<i>Law o excluded middle:</i>	$A \cup \bar{A} = X$
<i>Law of contradiction:</i>	$A \cap \bar{A} = \emptyset$
<i>DeMorgan laws:</i>	$\overline{A \cap B} = \bar{A} \cup \bar{B}$ $\overline{A \cup B} = \bar{A} \cap \bar{B}$

So far we have used the universal set X when we evaluated the complement of a given set. However, it is frequently of interest to find elements that belong to one set but do not belong to the other. Such a *relative complementation* of one set relative to another set is referred to as *subtraction* and is formally defined as follows:

$$A \setminus B = \{x \mid x \in A \text{ and } x \notin B\}$$

Of course, if A is the universal set X the above subtraction operation yields a complement of B , i.e. \bar{B} .

The three operations of union, intersection and subtraction of sets A and B are illustrated in Figure 1.

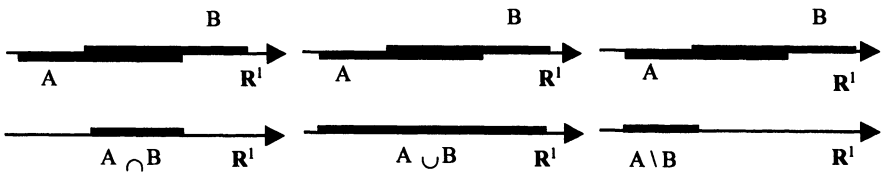


Figure 1. Intersection, union and subtraction operations on sets A and B .

Note that the operands and the results of the above operations are all sets in R . The *Cartesian product* and the *projection* operations produce sets that are qualitatively

different from those represented by the operands. The Cartesian product results in set elements that have dimension equal to the sum of dimensions of the operands while the projection operation operates in reverse and 'flattens' the set elements to the space defined by the projection set. These two operations are defined as follows:

Cartesian product of A and B is

$$A \times B = \{(x, y) \mid x \in A \text{ and } y \in B\}$$

Projection of a set C onto set A is

$$\text{proj}_A(C) = \{x \in A \mid \exists y \text{ for which } (x, y) \in C\}$$

It follows from the above definition that if C is a Cartesian product of A and B a projection of C onto A is a set A and a projection of C onto B is a set B. However, for an arbitrary set C, defined in the same space as the Cartesian product of two other sets A and B, the Cartesian product of the projections of this set onto sets A and B respectively is not equal to the set C. Indeed, $C \subset \text{proj}_A(C) \times \text{proj}_B(C)$ but not vice-versa. This is illustrated in Figure 2.

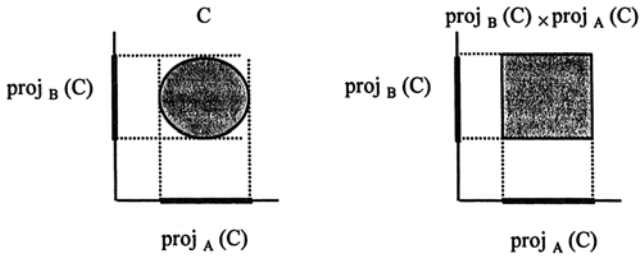


Figure 2. Cartesian product and projection operation.

Functional Mapping of Sets

The projection operation is a special case of a more general concept of functional mapping of one set into another set. We will denote the mapping function f that has elements of set A as operands and generates values that are elements of set B as $f: A \rightarrow B$. So, for every subset $A_1 \subset A$ we can write

$$f(A_1) = \{f(x) \mid x \in A_1\} = B_1$$

meaning that the function acts on all elements of the set A_1 and produces a set B_1 as a result. The reciprocal function $f^{-1}: B \rightarrow A$ that maps the image set B onto the set A is defined as follows

$$f^{-1}(B_1) = \{x \in A \mid f(x) \in B_1\}$$

where, the operand B_1 represents any subset $B_1 \subset B$.

It is worth noting that, with this definition of the inverse mapping, the superposition $f^{-1}(f(A_1))$ does not result in A_1 since it is possible for some $x \notin A_1$ to have an image $f(x)$ that is identical to the image $f(x)$ for some other $x \in A_1$. Therefore, we can only say that

$$A_1 \subset f^{-1}(f(A_1))$$

and conversely

$$f(f^{-1}(B_1)) \subset B_1$$

This is illustrated in Figure 3.

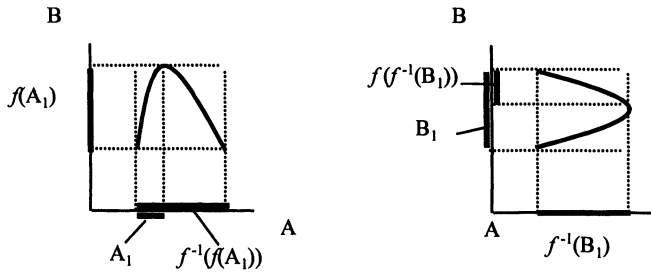


Figure 3. Composition of an image and a reciprocal image of a set.

It is easy to show that if A_1 and A_2 are subsets of A , the functional mapping of a union and intersection of A_1 and A_2 is equal to the union and intersection of functional mappings $f(A_1)$ and $f(A_2)$. Similarly, if B_1 and B_2 are subsets of B the inverse image of a union and intersection of B_1 and B_2 is equal to the union and intersection of inverse images $f^{-1}(B_1)$ and $f^{-1}(B_2)$. Furthermore, both the functional mapping and its inverse satisfy the monotonicity property. That is, if set A_1 is included in set A_2 then the image set $f(A_1)$ is also included in the image set $f(A_2)$ and if set B_1 is included in set B_2 then the inverse image set $f^{-1}(B_1)$ is included in the inverse image set $f^{-1}(B_2)$. This is summarized below using formal notation.

$$\begin{aligned} f(A_1 \cap A_2) &= f(A_1) \cap f(A_2) \\ f(A_1 \cup A_2) &= f(A_1) \cup f(A_2) \\ f^{-1}(B_1 \cap B_2) &= f^{-1}(B_1) \cap f^{-1}(B_2) \\ f^{-1}(B_1 \cup B_2) &= f^{-1}(B_1) \cup f^{-1}(B_2) \\ \text{if } A_1 &\subset A_2 \text{ then } f(A_1) \subset f(A_2) \\ \text{if } B_1 &\subset B_2 \text{ then } f^{-1}(B_1) \subset f^{-1}(B_2) \end{aligned}$$

Arithmetical Operations on Sets

The functional mapping of sets discussed so far (defined for all subsets A_i of a given source set A and resulting in subsets B_i of an image set B) can be regarded as a mapping of the power set $\mathcal{P}(A)$ into the power set $\mathcal{P}(B)$. Using this framework we can now consider the generalization of arithmetical operations on sets. Since the two operands in such operations are subsets selected from the corresponding operand domain sets, such generalized arithmetical operations can be regarded as mappings from the power set of the Cartesian product of the operand sets into a power set of the results set. To put it formally, if A and B represent operand sets and C represents a results set, the generalized arithmetical operation op can be regarded as a mapping from $\mathcal{P}(A \times B)$ into $\mathcal{P}(C)$. The semantics of the generalized op operation is defined using a standard version of op applied to individual elements of operand sets:

$$A_1 \text{ op } B_1 = \{ x_1 \text{ op } y_1 \mid x_1 \in A_1, y_1 \in B_1 \}$$

If, for instance, A_1 and B_1 are subsets of \mathbf{R}^n the “-” operation for sets evaluates as

$$A_1 - B_1 = \{ x - y \mid x \in A_1, y \in B_1 \}$$

However, it should be remembered that even if the two operands are identical sets, the result is not necessarily equal to $\{0\}$

$$A_1 - A_1 = \{ x - y \mid x \in A_1, y \in A_1 \} \neq \{0\}$$

This is because the generalized set operation “-” is being applied to all possible pairs x, y . So, even if x and y are drawn from the same set A_1 , it is only a special case when $x=y$. The consequence of processing all possible pairs of operands by the generalized set operations op is that the result set often overestimates the set of feasible results that obtained by applying op to individual elements of the operand sets. The degree of overestimation of the image set is dependent on the structure of computations performed on sets. In particular, the presence of terms $A_i - A_i$ gives rise to additional elements in the image set. This is called a *dependency effect* (Jaulin et al., 2001). We shall elaborate on the specific manifestations of the dependency effect in subsequent sections.

2. 3 SET ENCLOSURE

The presence of *dependency effect* prompts an interest in quantifying the overestimate of the image set. However, it is clear that direct enumeration of additional elements in the results set is unlikely to succeed in any but the simplest scenarios due to the cardinality and the topological complexity of the sets involved.

This leads to the consideration of set enclosures, which are structurally simple constructs that allow efficient computation of outer approximations of sets. In this book we will largely confine ourselves to hyperboxes in \mathbb{R}^n as enclosures of sets. Other enclosures such as ellipsoids, convex polyhedrons, etc. can be used as alternatives and have been reported in the literature (Milanese, 1996). However, the use of hyperboxes has a distinct advantage of supporting an intuitively straightforward interpretation of resulting sets. The price that is paid for the convenience of dealing with topologically simpler enclosures (in particular with hyperboxes) is the further overestimation of the operand and results sets. We emphasize here that this overestimation comes on top of the previously mentioned dependency effect. We shall refer to the overestimation due to the use of set enclosures as *wrapping effect*.

We can now extend the definition of the operator op , from mapping the elements of the power set $\mathcal{P}(A \times B)$ to $\mathcal{P}(C)$, to mapping between set enclosures. Before we do that however we need to introduce a couple of definitions. An *enclosure of a set* is defined as a smallest hyperbox containing such a set. We will denote such an enclosure by an uppercase letter in square brackets, e.g. $[A]$. If we now consider all possible subsets of the set A and denote them as A_i , we can say that sets A_i are elements of the power set $\mathcal{P}(A)$. So the enclosures $[A_i]$ are elements of the power set $\mathcal{P}([A])$. However, by definition, the power set $\mathcal{P}([A])$ contains all possible subsets of $[A]$ not just the hyperboxes. It is therefore convenient to define a *hyperbox power set*, as a set containing all possible hyperbox subsets of a given set enclosure. We shall denote the hyperbox power set as $\mathcal{H}([A])$. Clearly, $\mathcal{H}([A])$ is a subset of $\mathcal{P}([A])$, i.e. $\mathcal{H}([A]) \subset \mathcal{P}([A])$, but we can still assure that all enclosures $[A_i]$ belong to the hyperbox power set $\mathcal{H}([A])$.

If we now consider an operator op acting on specific sets A_1 and B_1 , that are subsets of the corresponding sets A and B , the result of applying this operator will, in general be a set that have a complex topology even if the operands are hyperboxes. Consequently we are often interested in the *enclosure of the operator op* , denoted as $[op]$, that is defined as an operator producing an enclosure of the results set obtained with operator op itself. So we can write

$$A_1 [op] B_1 = [\{ x_1 op y_1 \mid x_1 \in A_1, y_1 \in B_1 \}]$$

with the set enclosure operator $[.]$ defined as

$$[C_1] = \bigcap \{ D \in \mathcal{H}([C]) \mid C_1 \subset D \}$$

In other words, the enclosure $[C_1]$ of C_1 is the smallest element of $\mathcal{H}([C])$ that contains C_1 . So, by the above definition the enclosure operator $[op]$ always generates a set that contains the set generated by the operator op itself.

$$A_1 [op] B_1 \supset A_1 op B_1$$

Similarly we can define enclosures for functions. Consider sets A and B and their corresponding enclosure sets $\mathcal{A}([A])$ and $\mathcal{A}([B])$. The function $f: A \rightarrow B$ can be extended to enclosures for a specific $A_1 \subset \mathcal{A}([A])$ as follows

$$[f](A_1) = \{f(x) \mid x \in A_1\}$$

Again, it is clear that

$$[f](A_1) \supset f(A_1)$$

We illustrate, in Figure 4, the wrapping effects introduced by the function enclosure and the function acting on a set enclosure. The two, are in general quite different.

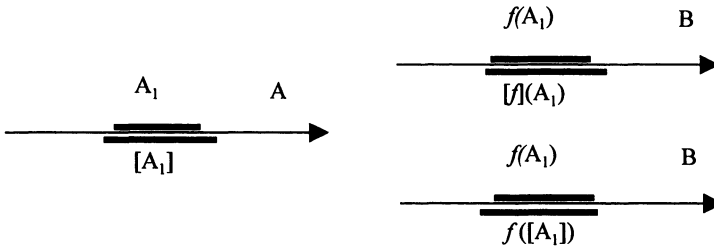


Figure 4. Overestimation of the function image set due to enclosure operation.

Note that the function enclosure guarantees that $[f](A_1) \supset f(A_1)$
and the set enclosure guarantees that $f([A_1]) \supset f(A_1)$ but
 $[f](A_1)$ is in general quite different from $f([A_1])$.

2. 4 INTERVAL ANALYSIS

Intervals are connected subsets in \mathbf{R} , so the *interval analysis* can be considered as a special case of set analysis. Although the real-life situations involve sets that have complex topologies, the intervals (or Cartesian product of intervals, i.e. hyperboxes) offer a convenient approximation of such sets because of the ease of manipulation of intervals and their straightforward interpretation.

Basic Interval Operations

Throughout this book we shall denote intervals by lowercase letters enclosed in square brackets, e.g. $[x]$. The intervals can be characterized unambiguously by their lower and upper bounds. The lower bound of the interval $[x]$ will be denoted by x^- and the upper bound by x^+ . More formally the lower bound can be defined as

$$x^- = \sup\{a \in \mathbf{R} \mid \forall x \in [x], a \leq x\}$$

and the upper bound

$$x^+ = \inf\{a \in \mathbf{R} \mid \forall x \in [x], x \leq a\}$$

Note that the lower and upper bounds defined as above may or may not belong to the interval $[x]$. In other words the interval $[x]$ may be closed or not. Using the x^- and x^+ bounds we can characterize the *width* and the *center* of the interval in a straightforward manner

$$\text{width}([x]) = x^+ - x^-$$

$$\text{center}([x]) = (x^+ + x^-)/2 = x^- + \text{width}([x])/2$$

The set theoretic operations introduced in the previous section can be applied to intervals. The *union* and *intersection* of two intervals $[x]$ and $[y]$ are defined as

$$[x] \cup [y] = \{a \in \mathbf{R} \mid a \in [x] \text{ or } a \in [y]\}$$

$$[x] \cap [y] = \{a \in \mathbf{R} \mid a \in [x] \text{ and } a \in [y]\}$$

It is clear that the intersection of intervals is always an interval (admittedly it could be an empty one). However, the union of intervals, as defined above, is not necessarily a connected subset in \mathbf{R} . So, in order to make intervals closed with respect to union operation we define an *interval union* that produces the smallest interval that contains $[x] \cup [y]$. We shall denote this operation as

$$[x] [\cup] [y] = [\min(x^-, y^-), \max(x^+, y^+)]$$

and can express it in terms of lower and upper bounds of the intervals as

$$[x] [\cup] [y] = [\min(x^-, y^-), \max(x^+, y^+)]$$

A similar comment applies to the subtraction operation, so we introduce *interval subtraction* as follows

$$[x] [\setminus] [y] = [x] \setminus [y] = [a, b]$$

where

$$a = \sup\{c \in \mathbf{R} \mid \forall d \in [x] \setminus [y], c \leq d\}$$

and

$$b = \inf\{c \in \mathbf{R} \mid \forall d \in [x] \setminus [y], d \leq c\}$$

The *minimum* and *maximum* operations are defined in terms of the lower and upper bounds of intervals as follows

$$\max([x],[y])=[\max(x^-,y^-), \max(x^+,y^+)]$$

and

$$\min([x],[y])=[\min(x^-,y^-), \min(x^+,y^+)]$$

The *projection* of interval $[x]$ onto \mathbf{R} is the interval $[x]$ itself and the *Cartesian product* of intervals $[x] \times [y]$ is a box in \mathbf{R}^2 . The former is of little practical use and the latter will be discussed in the following section. The selected operations on intervals of union, intersection and subtraction are illustrated in Figure 5. It is worth noting here that the interval union and subtraction are specific cases of operator enclosure introduced in the general context of sets in the previous section.

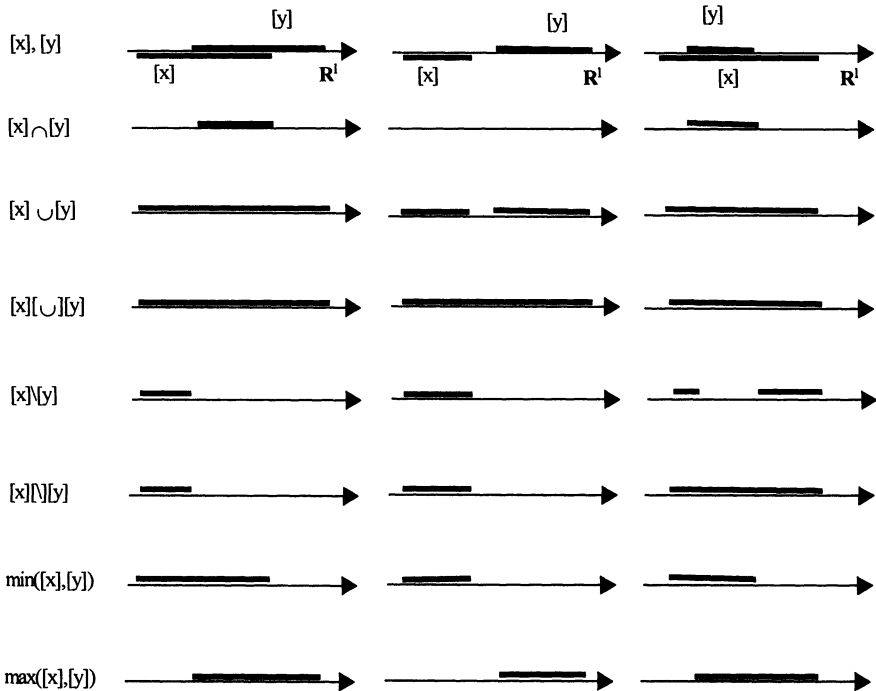


Figure 5. Operations on intervals.

Arithmetical Operations on Intervals

The above operations are valid for both closed and open intervals. However, for the non-empty, closed intervals we can also define four arithmetical operations since, in this case, the lower and upper bound belongs to the interval.

- *Addition* of intervals $[x]$ and $[y]$ is
 $[x]+[y]=[x^-+y^-, x^++y^+]$
- *Subtraction* of intervals $[x]$ and $[y]$ is
 $[x]-[y]=[x^- - y^+, x^+ - y^-]$
- *Multiplication* of intervals $[x]$ and $[y]$ is
 $[x][y]=[\min\{x^-y^-, x^-y^+, x^+y^-, y^+x^+\}, \max\{x^-y^-, x^-y^+, x^+y^-, y^+x^+\}]$
- *Multiplication* of interval $[x]$ by a scalar w
 $w[x]=[wx^-, wx^+]$ if $w > 0$
 $w[x]=[wx^+, wx^-]$ if $w < 0$
- *Division* of intervals $[x]$ and $[y]$ is
 $[x]/[y]=[x^-](1/[y])$

where, $1/[y] = [1/y^+, 1/y^-]$ if $y^- > 0$ or $y^+ < 0$

A note of caution is due here. Although the operations on intervals, and in particular arithmetical operations on closed intervals, look as straightforward extensions of arithmetical operations on scalar variables, these are in essence operations on sets. This means that the *dependency* effect identified for sets, operates also here. We can illustrate this through an example.

Example 1

Consider an interval $[x]=[-1, 2]$ for which we need to evaluate value of a function $[x][x]-2[x]$. By making a direct substitution for $[x]$ and applying the interval arithmetic rules we obtain

$$[x][x] - 2[x] = [-1, 2][-1, 2] - [-2, 4] = [-2, 4] - [-2, 4] = [-6, 6]$$

However we can also write the above function using only two occurrences of $[x]$. In this case the function evaluates as

$$[x]^2 - 2[x] = [0, 4] - [-2, 4] = [-4, 6]$$

Going one step further we can avoid the repetition of $[x]$ altogether and write

$$([x] - 1)^2 - 1 = [-2, 1]^2 - 1 = [0, 4] - [1, 1] = [-1, 3]$$

It is clear that the dependency effect can severely deteriorate the quality of the results by producing very conservative enclosures. Figure 6 illustrates this effect for the above three alternative forms. As a general rule it is recommended to transform the interval expression so as to avoid the repetition of the same interval. However, in practice, this can be quite difficult to achieve.

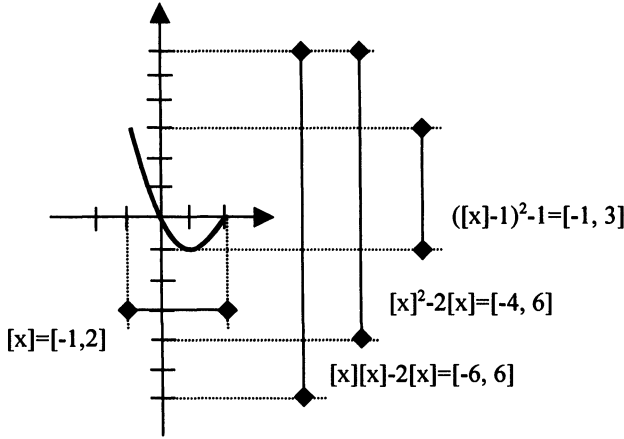


Figure 6. Dependency effect for operations on intervals.

Example 2

To assess the *wrapping effect* we can continue with the above example and consider the interval $[x] = [-1, 2]$ as an enclosure of the actual argument set $[a] = [-0.9, 1.9]$. Repeating the calculations for $[a]$ we have

$$[a][a] - 2[a] = [-1.71, 3.61] - [-1.8, 3.8] = [-5.51, 5.41]$$

$$[a]^2 - 2[a] = [0, 3.61] - [-1.8, 3.8] = [-3.8, 5.41]$$

$$([a] - 1)^2 - 1 = [-1.9, 0.9]^2 - 1 = [0, 3.61] - [1, 1] = [-1, 2.61]$$

As expected we can see that for the actual argument set $[a]$ the corresponding result sets are subsets of the result sets obtained for the argument enclosure set $[x]$; i.e.:

$$[-5.51, 5.41] \subset [-6, 6]$$

$$[-3.8, 5.41] \subset [-4, 6]$$

$$[-1, 2.61] \subset [-1, 3]$$

However, it is also clear that the dependency effect demonstrates itself in the same way regardless whether we deal with the argument set $[a]$ or the argument enclosure set $[x]$.

2.5 INTERVAL VECTORS

An *interval vector* $[x]$ in \mathbf{R}^n is defined as a Cartesian product of n intervals. We distinguish it from intervals $[x]$ in \mathbf{R} by using a boldface font.

$$[x] = [x_1] \times [x_2] \times \dots \times [x_n],$$

where $[x_i]$ are intervals, $[x_i] = [x_i^-, x_i^+]$, $i=1, \dots, n$

The above definition implies that $[x]$ is a specific subset in \mathbf{R}^n that has axis-aligned edges, i.e. it is a hyperbox. Using the notation from Section 3 we can write that $[x]$ is an element of a hyperbox power set $\mathcal{H}(\mathbf{R}^n)$, i.e. $[x] \in \mathcal{H}(\mathbf{R}^n)$. Figure 7 illustrates a two-dimensional hyperbox.

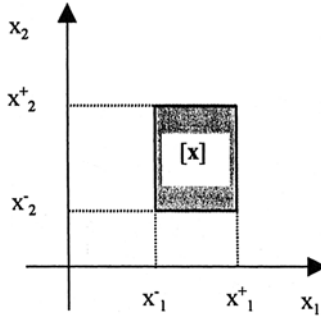


Figure 7. Two-dimensional hyperbox, $[x] \in \mathcal{H}(\mathbf{R}^2)$.

The characterization of hyperboxes is analogous to that introduced for intervals in \mathbf{R} . The lower left-hand corner of a hyperbox is referred to as a lower bound of the vector interval x^- and the top right-hand corner is referred to as an upper bound, x^+ . These can be expressed in terms of bounds on the component intervals

$$x^- = [x_1^-, x_2^-, \dots, x_n^-]^T$$

$$x^+ = [x_1^+, x_2^+, \dots, x_n^+]^T$$

However, while in a one-dimensional case the interval was fully characterized by its width and center, in case of hyperboxes these measures do not capture the *shape* of the vector interval in \mathbb{R}^n . It is useful therefore to consider a volume of a hyperbox as a means of identifying those special cases where one or more of the n projections of a hyperbox is a single point, i.e. $x_i^- = x_i^+$.

Using the bounds on the component intervals of a hyperbox we can characterize the *width*, *center* and the *volume* in a straightforward manner

$$\text{width}([x]) = \max_{i=1,\dots,n} (\text{width}([x_i]))$$

$$\text{center}([x]) = (\text{center}([x_1]), \text{center}([x_2]), \dots, \text{center}([x_n]))^T$$

$$\text{volume}([x]) = \text{width}([x_1]) * \text{width}([x_2]) * \dots * \text{width}([x_n])$$

The set theoretic operations of *union* and *intersection* of hyperboxes are defined here as generalizations of the corresponding operations on intervals.

$$[x] \cap [y] = ([x_1] \cap [y_1] \times [x_2] \cap [y_2] \times \dots \times [x_n] \cap [y_n])^T$$

$$[x] \cup [y] = ([x_1] \cup [y_1] \times [x_2] \cup [y_2] \times \dots \times [x_n] \cup [y_n])^T$$

Intersection of hyperboxes $[x]$ and $[y]$ yields always either a hyperbox or an empty set. However, the union typically produces a more complex set except for two special cases; when:

- i. one of the hyperboxes is contained in the other, i.e. $[x] \subset [y]$ or $[x] \supset [y]$;
- ii. all but one of the component intervals of the two hyperboxes are identical, i.e. $[x_i] = [y_i]$, for $i=1, \dots, n$ and $i \neq j$.

A typical case of a union of hyperboxes and a special case where the hyperboxes differ only in one dimension is illustrated in Figure 8.

In order to make hyperboxes closed with respect to the union operation we define a *hyperbox union* that produces the smallest hyperbox that contains $[x] \cup [y]$. We shall denote this operation as $[\cup]$

$$[x] [\cup] [y] = [[x] \cup [y]]$$

An analogous concern about topologically complex (and possibly disjoint) result of the subtraction operation requires introduction of a *hyperbox subtraction* as follows

$$[x] [\setminus] [y] = [[x] \setminus [y]]$$

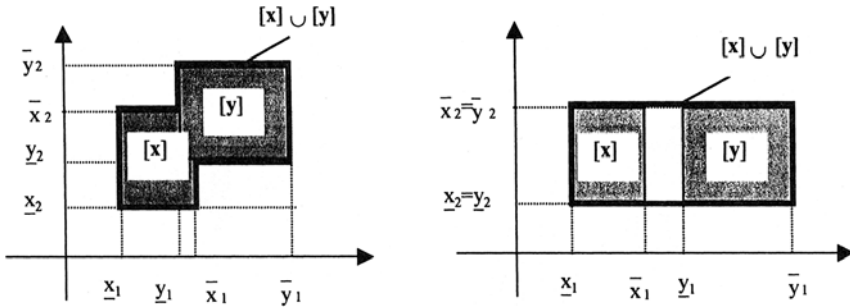


Figure 8. A typical and a special case of a union of hyperboxes.

Clearly the above operations extend to any number of hyperboxes. The arithmetical operations are also a direct extension of the same operations on intervals. For instance, if $[x]$ and $[y]$ are elements of $\mathcal{H}(\mathbf{R}^n)$, and w is a scalar $w \in \mathbf{R}$, then we can define the *product of a scalar and a hyperbox*, the *product of two hyperboxes* and the *sum of hyperboxes* as follows

$$w[x] = (w[x_1]) \times (w[x_2]) \times \dots \times (w[x_n])$$

$$[x]^T[y] = [x_1][y_1] + [x_2][y_2] + \dots + [x_n][y_n]$$

$$[x]^T + [y] = ([x_1] + [y_1]) \times ([x_2] + [y_2]) \times \dots \times ([x_n] + [y_n])$$

Note that the $w[\cdot]$, $[\cdot][\cdot]$, $[\cdot] + [\cdot]$ and $[\cdot] \times [\cdot]$ operations are as defined in Section 4.

2. 6 INTERVAL MATRICES

The hyperboxes discussed so far were interpreted as interval vectors that have dimension $1 \times n$. It is a straightforward thing to generalize this interpretation to interval matrices of size $m \times n$. Please note that this generalization does not affect the interpretation of hyperboxes as sets; this time in \mathbf{R}^{mn} , or more precisely as elements of $\mathcal{H}(\mathbf{R}^{mn})$.

An *interval matrix* $[A]$ in \mathbf{R}^{mn} is defined as a Cartesian product of ‘mn’ intervals or alternatively as a Cartesian product of n , m -dimensional interval vectors.

$$[A] = \begin{bmatrix} [a_{11}] & \dots & [a_{1n}] \\ \dots & \dots & \dots \\ [a_{m1}] & \dots & [a_{mn}] \end{bmatrix} = [a_1] \times \dots \times [a_n]$$

where $[a_j]$ are interval vectors, $[a_j] = [a_{1j}] \times \dots \times [a_{mj}]$, $j=1, \dots, n$.

Each interval element of the matrix $[A]$ is uniquely characterized by its upper and lower bound, a_{ij}^+ and a_{ij}^- respectively, so the upper bound on the interval matrix $[A]$ is defined as a matrix composed of upper bounds of the constituent intervals. This is denoted by A^+ . Similarly the lower bound of the interval matrix $[A]$ is defined as a matrix composed of lower bounds of the constituent intervals and is denoted by A^- . It is worth noting that both A^+ and A^- are no longer interval matrices but simply numerical matrices in \mathbf{R}^{mn} .

The characterization of interval matrices in terms of *width*, *center* and *volume* follows the definitions provided for interval vectors in the previous section.

$$\text{width}([A]) = \max_{j=1, \dots, n} (\text{width}([a_j]))$$

$$\text{center}([A]) = (\text{center}([a_1]), \text{center}([a_2]), \dots, \text{center}([a_n]))$$

$$\text{volume}([A]) = \text{width}([a_1]) * \text{width}([a_2]) * \dots * \text{width}([a_n])$$

The set theoretic operations of *union*, *intersection* and *subtraction* of interval matrices are defined here as generalizations of the corresponding operations on interval vectors.

$$[A] \cap [B] = ([a_1] \cap [b_1] \times [a_2] \cap [b_2] \times \dots \times [a_n] \cap [b_n])$$

$$[A] \cup [B] = ([a_1] \cup [b_1] \times [a_n] \cup [b_n] \times \dots \times [a_n] \cup [b_n])$$

$$[A] \setminus [B] = ([a_1] \setminus [b_1] \times [a_n] \setminus [b_n] \times \dots \times [a_n] \setminus [b_n])$$

Bearing in mind the properties of these operations on interval vectors, as discussed in Section 5, it is clear that intersection of interval matrices $[A]$ and $[B]$ will yield either a hyperbox or an empty set but the union and subtraction operations will typically produce sets that are not elements of $\mathcal{H}(\mathbf{R}^{mn})$. We need therefore hyperbox versions of these operations to ensure that the results are hyperboxes in \mathbf{R}^{mn} .

We define a hyperbox union of interval matrices as

$$[A] [\cup] [B] = [[A] \cup [B]]$$

and a *hyperbox subtraction* as

$$[A] [\setminus] [B] = [[A] \setminus [B]]$$

The arithmetical operations on interval matrices are a direct consequence of operations on interval vectors. For instance, if $[A]$ and $[B]$ are $n \times n$ interval matrices, $[x]$ is an n -dimensional interval vector and w is a scalar we have

$$w[A] = (w[a_1]) \times (w[a_2]) \times \dots \times (w[a_n])$$

$$[A][B] = ([a_i]^T [b_j]), 1 \leq i \leq n, 1 \leq j \leq n$$

$$[A][x] = ([a_i]^T [x]), 1 \leq i \leq n$$

$$[A] + [B] = ([a_i] + [b_i]), 1 \leq i \leq n$$

Unfortunately, some of the properties that are pertinent to operations on numerical matrices no longer hold for interval matrices. For instance, the arithmetical product of interval matrices is no longer associative or distributive, i.e.

$$([A][B])[C] \neq [A]([B][C])$$

$$[A]([B] + [C]) \neq ([A][B] + [A][C])$$

However, the interval matrices satisfy the so-called *subdistributivity* property, which stipulates that

$$[A]([B] + [C]) \subset ([A][B] + [A][C])$$

We notice that this is consistent with the *dependency* effect, discussed earlier on, that manifested itself when there was a repetition of an interval variable in the arithmetical expression. The violation of associativity is, on the other hand, due to the *wrapping* effect inherent to multiplication of interval matrices.

Example 3

Let's consider three interval matrices $[A] = \begin{bmatrix} [-1,1] & [-2,-1] \\ [0,0] & [3,3] \end{bmatrix}$, $[B] = \begin{bmatrix} [-1,1] & [1,2] \\ [0,0] & [2,3] \end{bmatrix}$ and $[C] = \begin{bmatrix} [-3,-2] & [0,1] \\ [-2,-1] & [-1,2] \end{bmatrix}$.

Checking the *associativity* we have

$$([A][B])[C] = \begin{bmatrix} [-1,1] & [-8,0] \\ [0,0] & [6,9] \end{bmatrix} \begin{bmatrix} [-3,-2] & [0,1] \\ [-2,-1] & [-1,2] \end{bmatrix} = \begin{bmatrix} [-3,19] & [-17,1] \\ [-18,-6] & [0,18] \end{bmatrix}$$

which is different from

$$[A]([B][C]) = \begin{bmatrix} [-1,1] & [-2,-1] \\ [0,0] & [3,3] \end{bmatrix} \begin{bmatrix} [-7,2] & [-1,5] \\ [-6,-2] & [0,6] \end{bmatrix} = \begin{bmatrix} [-5,19] & [-17,5] \\ [-18,-6] & [0,18] \end{bmatrix}$$

And checking the *distributivity* we obtain

$$[A]([B]+[C]) = \begin{bmatrix} [-1,1] & [-2,-1] \\ [0,0] & [3,3] \end{bmatrix} \begin{bmatrix} [-4,-1] & [1,3] \\ [-2,-1] & [2,5] \end{bmatrix} = \begin{bmatrix} [-3,8] & [-11,-1] \\ [-6,-3] & [6,15] \end{bmatrix}$$

which is again different from

$$[A][B]+[A][C] = \begin{bmatrix} [-1,1] & [-8,0] \\ [0,0] & [6,9] \end{bmatrix} + \begin{bmatrix} [-2,7] & [-5,1] \\ [-6,-3] & [0,6] \end{bmatrix} = \begin{bmatrix} [-3,8] & [-13,1] \\ [-6,-3] & [6,15] \end{bmatrix}$$

Example 4

To illustrate the wrapping effect we consider a product of an interval matrix $[A]$ and an interval vector $[x]$. The numerical values are $[A] = \begin{bmatrix} [-1,1] & [-2,-1] \\ [0,0] & [3,3] \end{bmatrix}$, $[x] = \begin{bmatrix} [1,2] \\ [2,3] \end{bmatrix}$.

The result of multiplication of $[A][x]$ is the interval vector (hyperbox) $[b]$

$$[b] = \begin{bmatrix} [-8,0] \\ [6,9] \end{bmatrix}$$

which represents the enclosure set for the actual results set $B = \{[A]x \mid x \in [x]\}$. This is illustrated in Figure 9. Indeed, the hyperbox $[b]$ contains points that cannot be obtained for any point $x \in [x]$. For example, $x = [-8, 6]^T$ or $x = [0, 9]^T$.

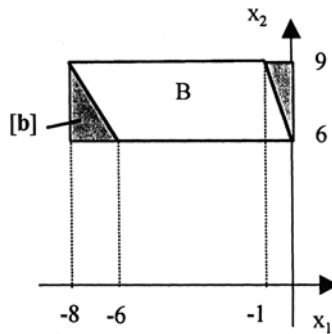


Figure 9. Wrapping effect inherent to multiplication of hyperboxes.

2. 7 ENCLOSURE OF FUNCTIONS

The interval computations discussed in Sections 4 to 6 give basis for the derivation of hyperbox enclosures for arbitrary functions f defined on interval $[x]$. Given that any function can be expressed as a sequence of elementary arithmetical operations, we can find a hyperbox enclosure of the results set by replacing the arithmetical operations with the equivalent interval arithmetic. This is referred to as a *natural enclosure* of the function $f([x])$ and is denoted as $[f]([x])$. However, the natural enclosure is often too wide. What is actually needed is the smallest enclosure possible that minimizes the overestimation of the results set. We shall denote such an enclosure as $[f]^*([x])$. In principle, finding $[f]^*([x])$ can be expressed as two optimization problems; one dealing with finding an *infimum* of function $f([x])$ and the other with finding a *supremum*. However, these optimization problems are frequently quite difficult to solve.

Example 5

To illustrate the point we consider the following function (see Figure 10)

$$f(x) = x - x^3/6 + x^5/120, \quad x \in [-3, 3]$$

The natural enclosure of this function $[f]([x])$ is as follows

$$[f]([x]) = [x] - [x]^3/6 + [x]^5/120 = [-3, 3] - [-4.5, 4.5] + [-2.025, 2.025]$$

i.e. it produces an interval $[9.525, -9.525]$ as an enclosure for the results set. This is a vast overestimate for the results set that is approximately contained within the $[-1, 1]$ interval.

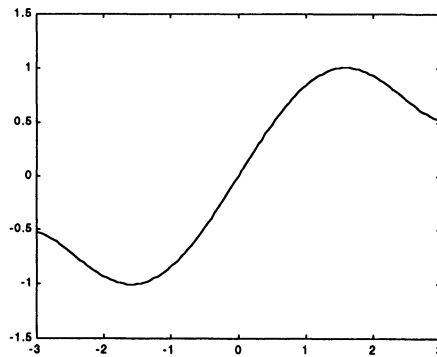


Figure 10. Function $f(x) = x - x^3/6 + x^5/120$.

The evaluation of the smallest enclosure $[f]^*$ can be accomplished here through finding a minimum and maximum of f . We calculate f' and solve the resulting equation $1 - x^2/2 + x^4/24=0$. This yields four roots $x=-1.592$, $x=1.592$, $x=-3.07$ and $x=3.07$. Since the last two roots lie outside the interval $[-3, 3]$ we calculate only the values $f(-1.592)$ and $f(1.592)$, which gives the smallest enclosure $[f]^*=[-1.005, 1.005]$. This is of course a significant improvement on the natural enclosure but we note, that for complex multi-dimensional functions this optimization can be a challenging task.

Centered Enclosures

One possible approach to improving on the tightness of the natural enclosure of f is to use the mean value theorem. Let $f: \mathbf{R}^n \rightarrow \mathbf{R}$ be a function of \mathbf{x} that is differentiable over the interval vector $[\mathbf{x}]$. We denote the center of this interval as $\mathbf{c}=\text{center}([\mathbf{x}])$ and its width in all dimensions as $\mathbf{w}=[\text{width}([x_1]), \dots, \text{width}([x_n])]$. The mean value theorem implies that

$$f(\mathbf{x})=f(\mathbf{c})+f'(\mathbf{x})(\mathbf{x}-\mathbf{c});$$

where $f'(\cdot)$ is the gradient of $f(\cdot)$ calculated with respect of individual variables x_i , $i=1, \dots, n$. Bearing in mind that \mathbf{c} represents a mid-point of the interval $[\mathbf{x}]$ we can conclude that $(\mathbf{x}-\mathbf{c}) \in [-(\mathbf{w}/2), \mathbf{w}/2]$ for every $\mathbf{x} \in [\mathbf{x}]$. So, the *centered enclosure* of the function $f(\cdot)$, denoted here as $[f_c](\mathbf{x})$, can be calculated as

$$[f_c](\mathbf{x})=f(\mathbf{c})+[f'](\mathbf{x})[-(\mathbf{w}/2), \mathbf{w}/2]$$

The $[f'](\mathbf{x})$ represents an enclosure of the gradient function. This enclosure should ideally be a minimum enclosure $[f']^*$ but, if that is not available, it can be either a centered enclosure (if the function is still differentiable) or a natural enclosure. Note that $f(\mathbf{c})$ is a scalar that provides an offset to the enclosing hyperbox calculated from the gradient and the width of the interval vector. A generalization of the centered enclosure for functions $f: \mathbf{R}^n \rightarrow \mathbf{R}^m$ is a straightforward combination of enclosures calculated for individual dimensions $j=1, \dots, m$.

Example 6

The calculation of the centered enclosure $[f_c]$, of the function illustrated in Figure 10, is accomplished as follows. As a first step we evaluate the enclosure of the gradient within the interval $[\mathbf{x}]$. We could calculate a natural-, centered- or a minimum enclosure of the gradient function depending on the required trade-off between the computational complexity and the tightness of bounding. In this example we choose to calculate the minimum enclosure $[f']^*$. This is accomplished by solving $f''(\mathbf{x})=0$ and finding a minimum and maximum value of $f'(\cdot)$ for the roots of $f''(\mathbf{x})=0$. Since $f''(\mathbf{x})=-\mathbf{x}+\mathbf{x}^3/6$, the roots are $\mathbf{x}=0$, $\mathbf{x}=-2.449$ and $\mathbf{x}=2.449$. So, the corresponding

values of the gradient function are $f'(0)=1$, $f'(2.449)=-0.5$ and $f'(-2.449)=-0.5$ thus giving the minimum enclosure $[f']^*([x])=[-0.5, 1]$.

Taking into account that the value of the function f in the center of the interval $[x]$ is $f(c)=0$ and the width of the interval is $\text{width}([x])=[-3, 3]$ we find that the centered enclosure of function f calculated with the minimum enclosure of the gradient f' is

$$[f_c]([x])=[-0.5, 1][{-3, 3}]=[{-3, 3}]$$

This is a significant improvement on the natural enclosure $[f]([x])=[-9.525, 9.525]$ (see Example 5 above). However, compared to the minimum enclosure of this function $[f]^*=[-1.005, 1.005]$ there is a significant room for improvement.

In order to motivate the refinement of the centered enclosure method we illustrate the above calculation in Figure 11.

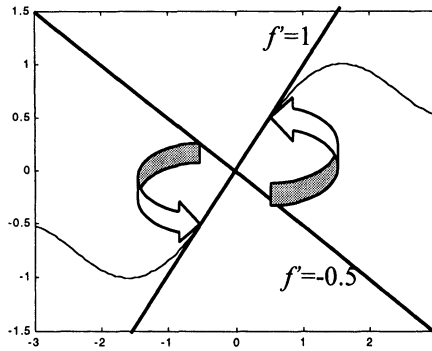


Figure 11. Centered enclosure of a function $[f_c]([x])=[-3, 3]$.

The gradient lines define the interval $[-3, 1.5]$ as possible values of the function to the left of the center $c=0$ and interval $[-1.5, 3]$ to the right of the center.

Space Subdivision Enclosures

We can see from the previous section that the width of the interval on which the function is evaluated has a direct influence on the size of hyperbox enclosure of the function. This is because the minimum and maximum gradient of the function is extrapolated to the whole of the interval $[x]$ even if these values are attained each for a single argument $x \in [x]$. We can therefore reduce the size of the function enclosure by subdividing the interval $[x]$ into smaller intervals $[x_i]$, $i=1, \dots, n$, such that $[x_1] \cup [x_2] \cup \dots \cup [x_n] = [x]$ and evaluating enclosures on the component intervals

$[x_i]$. The enclosure of the function defined on the whole interval $[x]$ is then taken as the union of the component enclosures.

$$[f]([x]) = [f]([x_1]) \cup [f]([x_2]) \cup \dots \cup [f]([x_n])$$

The calculation of function enclosures on the sub-intervals $[x_i]$ has the added benefit of identifying the local maximum (minimum) gradients, which are, by definition, not greater (smaller) than the maximum (minimum) gradient found on the whole of the interval $[x]$.

It is worth mentioning here that the idea of space subdivision underpins the granular view of the world that we discuss in Chapter xx. We can say that we are considering at first a low-resolution approximation (enclosure) of the original function and if this approximation is not adequate, we consider higher-resolution approximations. The choice of a specific subdivision of the interval $[x]$ corresponds to the choice of a specific resolution (level of granularity).

Example 7

We illustrate the operation of the space subdivision based calculation of function enclosure using the function utilized in Examples 5 and 6. We subdivide $[x] = [-3, 3]$ into three sub-intervals $[x_1] = [-3, -1]$, $[x_2] = [-1, 1]$ and $[x_3] = [1, 3]$. Calculating the minimum enclosures for the gradient on these sub-intervals we obtain:

$$\begin{aligned} [f']^*([x_1]) &= [-0.5, 0.54] \\ [f']^*([x_2]) &= [0.54, 1] \\ [f']^*([x_3]) &= [-0.5, 0.54] \end{aligned}$$

The centred enclosures of the function on sub-intervals are

$$\begin{aligned} [f_c]([x_1]) &= -0.93 + [-0.5, 0.54][-1, 1] = [-1.47, -0.39] \\ [f_c]([x_2]) &= [0.54, 1][-1, 1] = [-1, 1] \\ [f_c]([x_3]) &= 0.93 + [-0.5, 0.54][-1, 1] = [0.39, 1.47] \end{aligned}$$

So, the enclosure of the function on the whole $[x]$ is calculated as

$$[f]([x]) = [-1.47, -0.39] \cup [-1, 1] \cup [0.39, 1.47] = [-1.47, 1.47]$$

This is a significant improvement on the centred enclosure of the function f calculated on the whole interval $[x]$. The enclosure obtained with three subdivisions compares well to the minimum enclosure of $[-1.005, 1.005]$. However, a further improvement is easily attainable by increasing the number of subdivisions of $[x]$.

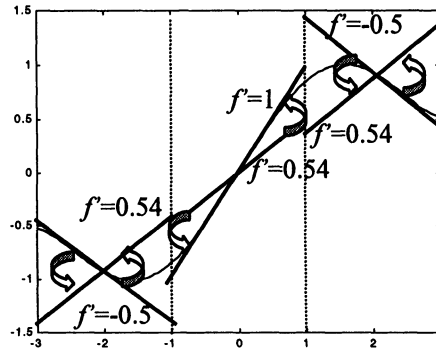


Figure 12. Centered enclosure of the function evaluated at first on sub-intervals and taken as a union of component enclosures.

The gradient lines define intervals enclosing the values of the function evaluated in individual sub-intervals.

These are: $[-1.47, -0.39]$, $[-1, 1]$ and $[0.39, 1.47]$.

2. 8 CONCLUSIONS

Sets provide a powerful framework for mathematical description of real-life objects. However, the generality of sets makes them rather difficult to use in computations. This is why we frequently confine ourselves to consideration of set enclosures, which are topologically simple sets that are easy to represent and manipulate by computers. In particular, we focus our attention on intervals and hyperboxes as convenient enclosures of sets.

The price that is paid for the interpretability of hyperbox enclosures is their inherent overestimation of sets. This is referred to as a *wrapping effect*. The wrapping effect applies to both the arguments and the results of set-valued functions so that the cumulative effect can be very significant. The focus of computations using set enclosures is therefore on the development of efficient techniques that minimize the overestimation of sets. One way to achieve that is to utilize the mean value theorem and to build enclosures around the center point of the argument set. Another powerful technique is to subdivide the space and to calculate the set enclosure as a union of individual enclosures in the subdivided space. The latter is frequently interpreted as multi-resolution processing (granulation) of information. The low-resolution analysis provides a quick, approximate solution and the high-resolution analysis improves on that result at the expense of greater computational effort.

Another factor contributing to the overestimation of results in set computations is the so-called dependency effect. The dependency effect arises when there are repeating occurrences of set variables in functional expressions. This effect is inherent to computations with sets in general and is not confined to computations using set enclosures. The overestimation due to dependency effect can be reduced by transforming the formal expression or function so as to decrease the number of occurrences of individual variables. However, such a transformation can be difficult to accomplish in many practical situations.

REFERENCES

- Ackermann, W., (1956), Zur axiomatische der mengenlehre, *Mathematische Annalen*, 131, 336-345.
- Bargiela A., Hainsworth G., (1989), Pressure and flow uncertainty in water systems, *ASCE Journal of Water Resources Planning and Management*, 115(2), 212-229.
- Bargiela, A. (2001) Interval and ellipsoidal uncertainty in water system state estimation, in: *Granular Computing*, (Pedrycz, W., ed.), Physica Verlag, 23-57.
- Barwise, J., (1989), Situated set theory, In *The Situation in Logic*, Center for the study of Language and Information, Stanford, CA, 289-292.
- Bourbaki, N., (1948), L'architecture des mathematiques, *Les Grands Courants de la Pensee Mathematique*, F. Le Lionnais (ed.), Cahiers du Sud.
- Fogel, E., Huang, Y.F., (1982), On the value of information in system identification – bounded noise case, *Automatica*, 18(2), 229-238.
- Goedel, K., (1940), *The consistency of the axiom of choice and of the generalised continuum hypothesis with the axioms of set theory*, Princeton University Press.
- Goedel, K., (1947), What is Cantor's continuum problem, *American Mathematical Monthly*, 54, 515-525.
- Hansen, E., (1975), A generalized interval arithmetic, *Lecture Notes in Computer Science*, Springer Verlag, Vol. 29, 7-18.
- Jaulin, L., Walter, E., (1983), Set inversion via interval analysis for nonlinear bounded-error estimation, *Automatica*, 29(4), 1053-1064.
- Jaulin, L., Kieffer, M., Didrit, O., Walter, E., (2001), *Applied Interval Analysis*, Springer, London
- Kearfott, R.B., Kreinovich, V., (eds.), (1996), *Applications of Interval Computations*, Kluwer, Dordrecht.
- Kreinovich, V., Lakeyev, A., Rohn, J., Kahl, P., (1998), *Computational Complexity and Feasibility of Data Processing and Interval Computations*, Kluwer, Dordrecht.
- Kreinovich, V., Wolff von Gudenberg, J., (1999), Arithmetic of complex sets: Nickel's classical paper revisited from a geometric viewpoint, *Geoinformatics*, Vol.9, 1, 21-26.

Mendelson, E., (1987), *Introduction to Mathematical Logic*, Wadsworth & Brooks/Cole, Belmont, CA.

Milaneze, M., Belforte, G., (1982), Estimation theory and uncertainty intervals evaluation in presence of unknown but bounded errors: Linear families of models and estimators, *Ieee Transactions on Automatic Control*, 27(2), 408-414.

Moore, R.E., (1959), Automatic error analysis in digital computation. TR Space Div. LMSD84821, Lockheed Missiles and Space Co.

Moore, R.E., (1962), Interval Arithmetic and Automatic Error Analysis in Digital Computing, *Ph.d. Dissertation, Department of Mathematics, Stanford University*, Stanford, California, Nov. 1962. Applied Mathematics and Statistics Laboratories TR-25.

Moore, R.E., (1965), The automatic analysis and control of error in digital computing based on the use of interval numbers. In L. B. Rall, (ed.), *Error in Digital Computation*, Vol. 1, 61-130. John Wiley and Sons, Inc.

Morse, A., (1965), *A Theory of Sets*, Academic Press, San Diego, CA.

Nickel, K., (1981), A globally convergent ball Newton method, *SIAM Journal on Numerical Analysis*, 18(6), 988-1003.

Shweppe, F.C., (1968), Recursive state estimation: unknown but bounded errors an system inputs, *IEEE Transactions on Automatic Control*, 13(1), 22-28.

Stadtherr, M.A., Schnepper, C.A., Brennecke, J.F., (1994), Robust phase stability analysis using interval methods, *International Symposium on Foundations of Computer Aided Process Design (FOCAPD'94)*.

Sunaga, T., (1958), Theory of Interval Algebra and its Applications to Numerical Analysis, *Gaukutsu Bunken Fukeyu-kai*, Tokyo.

Warmus, M., (1956), Calculus of approximations, *Bulletin de l'Academie Polonaise des Sciences*, 4(5), 253-259.

Warmus, M., (1961), Approximations of inequalities in the calculus of approximations: Classification of approximate numbers, *Bulletin de l'Academie Polonaise des Sciences*, 9(4), 241-245.

FUZZY SETS

As discussed in Chapter 2, sets or intervals are generic models of information granules. They dwell on a fundamental notion of dichotimization that bluntly states that any given element belongs to a certain concept or becomes excluded from it. The dichotomy is the underlying philosophical doctrine that goes back to Aristotle. While permeating logic and mathematics and being fully endorsed there, its validity and omnipresence has been challenged in different ways. This challenge arose through the pioneering developments of many valued logic led by Jan Lukasiewicz and Emil Post and a non-Aristotelian approach to philosophy of science promoted by Alfred Korzybski. While these were deeply rooted in the realm of philosophy and logic, the notion of fuzzy sets introduced by Lotfi Zadeh in 1965 has become highly appealing at the applied end of the spectrum of the challenges to the two-valued logic. As a consequence, they have found a vast array of applications and seriously questioned the very essence of the principle of dichotomy.

In this chapter, we introduce the idea of fuzzy sets, discuss their underlying fundamentals and elaborate on the essential processing schemes of such information granules.

3.1 THE CONCEPT AND FORMALISM

Fuzzy sets offer a possibility to formally express concepts of continuous boundaries. These concepts are everywhere and they are the core of our perception processes. When expressing ideas, describing concepts and communicating them to others, we always, we use terms to which the yes-no quantification does barely apply. Natural language is an evident environment of such communication. When forming and using concepts and communicating ideas and even simple observations, we use linguistic terms. The core issue there is that of building these underlying concepts (no matter in which area of human endeavors we are talking about). For instance, *low* inflation rate, *high* pressure, *small* approximation error, *medium* income are just few illustrative examples. Objects occurring in an image do not exhibit sharp boundaries. It becomes apparent that while these concepts are useful in the context of a certain problem as well as convenient in any communication realized in natural language, their set based formal model will lead us to a serious representation

drawback. We sense that determining a binary (yes-no, included-excluded) boundary between the elements that satisfy the term of *low inflation rate* is neither realistic nor formally sound. Our perception suggests that there could be (and are) some elements whose membership to the concept could be only partial. This problem was succinctly raised in the past when struggling with the better and more comprehensive understanding the nature of such perception problems. In Duhem (1906) we find the following observation

... it is impossible to describe a practical fact without attenuating by the use of the word "approximately" or "nearly": on the other hand all the elements constituting the theoretical fact are defined with rigorous exactness...

...this "mathematics of approximation" is not a simpler and cruder form of mathematics. On the contrary, it is a more thorough and more refined form of mathematics, requiring the solution of problems at time enormously difficult, sometimes even transcending the methods at the disposal of algebra today"

The observation about vagueness made by Russel is at the heart of a no-binary nature of expressions of natural language

...Vagueness and precision alike are characteristics which can only belong to a representation, of which language is an example. They have to do with relation between a representation and that which it represents. Apart from representation, whether cognitive or mechanical, there can be no such thing as vagueness or precision; things are what they are, and there is an end of it... The law of excluded middle is true when precise symbols are employed, but it is not true when symbols are vague, as, in fact, all symbols are... All traditional logic habitually assumes that precise symbols are being employed. It is therefore not applicable to this terrestrial life, but only to an imaginary celestial existence.

N. Wiener (1923) put is succinctly as

... experience only whispers "yes" or "no" in reply to our question, while logic shouts

A.Korzybski (1933) was the one who challenged the principle of dichotimization

..in analyzing the Aristotelian codification, I had to deal with the two-valued, "either-or" type of orientation. In living, many issues are not so sharp, and therefore a system that posits the general sharpness of "either-or" and so objectifies "kind" , is unduly limited; it must be revised and more flexible in terms of "degree"...

M. Black (1937) pointed at the discrepancy between the ideal objects being used in the formation of any theory and the relevance and suitability of such objects in real world

...it is a paradox, whose important familiarity fails to diminish, that the most highly developed and useful scientific theories are ostensibly expressed in terms of objects never encountered in experience. And the "point-planet" of astronomy, the "perfect gas" or thermodynamics or the "pure species" of genetics are equally remote from exact realization...

It was L.A. Zadeh who in his pioneering 1965 paper (Zadeh, 1965; Zadeh 1975; Zadeh 1978) coined a term of a fuzzy set as an object for which we admit partial membership of elements. Fuzzy set A is described by a membership function (usually denoted by $A(x)$ or $\mu_A(x)$) which maps the universe of discourse (X) in which A is defined into a unit interval

$$A : X \rightarrow [0,1] \quad (1)$$

Formally, $A(x)$ denotes a degree of membership that describes an extent to which x belongs to A . If $A(x) = 1$ then we say that x fully belongs to A . If $A(x)$ is equal to zero, x is fully excluded from A . The values of the membership function that are in-between 0 and 1 point at a partial membership of x to A . The higher the membership grade, the stronger the association of the given element to the concept.

Membership functions quantify the notion of partial membership. We can anticipate a diversity of the membership functions while aiming at the representation of the problem. Several typical membership functions are portrayed in Figure 1. Each of them can be treated as a convenient representation mechanism to deal with some class of descriptors (as again visualized in Figure 1). They are interesting tools to capture and quantify the form of transition (partial membership) that occurs in the description of the problem.

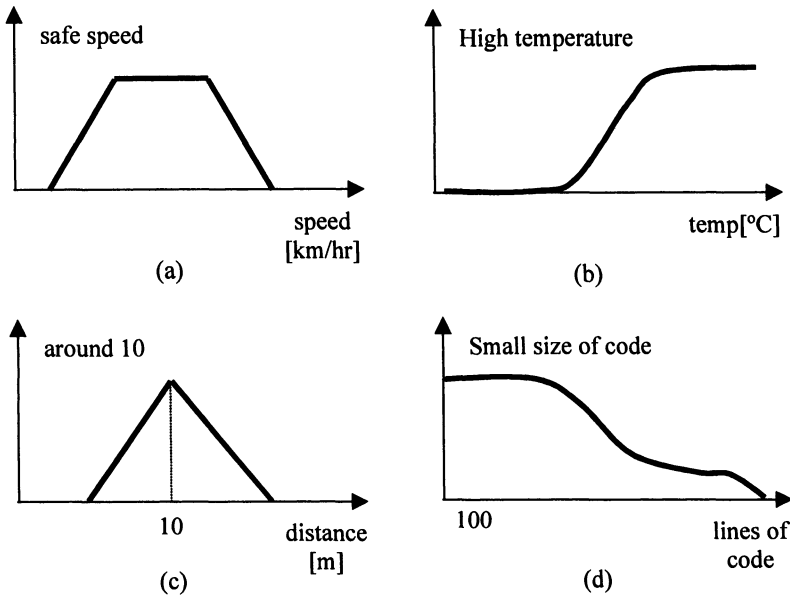


Figure 1. Classes of membership functions and their relation to some selected categories of concepts originating in a certain class of problems.

Dubois and Prade (1997) discussed three points of view at fuzzy sets (semantics of fuzzy sets); these points of view are closely linked with some general areas of applications in which membership grades take on a specific meaning that coincides with the specificity of the problem

Degree of similarity $A(x)$ is treated as a degree of proximity to prototype element(s) in A . This is an interpretation that is particularly appealing in the realm of pattern recognition (see also Bellman et al., 1966) where we are faced with some prototypical patterns (representatives). For instance, in fuzzy clustering, this point of view is predominant: we have a collection of prototypes being viewed as abstractions of all patterns. In the sequel, any new pattern is viewed vis-à-vis the existing prototypes (that usually coincide with the classes of objects identified in the problem) and compared how far it can be sought as a typical for each class. This helps identify patterns that are quite “typical” for the given class and flag those whose typicality is quite low and this may trigger a more thorough look at them.

Degree of preference A represents a collection of more or less preferred objects and $A(x)$ denotes a level of preference in favor of x or the feasibility of selecting x as the value in the decision process (Saaty, 1980). This point of view is deeply rooted in

the realm of decision analysis. In such setting approximate reasoning is essentially about the propagation of preferences through the system describing a certain model.

Degree of uncertainty This semantics is developed in the setting of the possibility theory where we view $A(x)$ as a degree of possibility that some variable assumes value x , given that all we know about it is that x “is A ”

As in case of sets, we will denote a family of fuzzy sets defined in a certain space X by $\mathcal{A}(X)$.

3. 2 THE DESCRIPTION AND GEOMETRY OF FUZZY SETS

Fuzzy sets are fully described by membership functions. From the practical standpoint, it is of interest to have a few descriptors of a fuzzy set that could be helpful in its concise characterization (Zimmermann, 2001; Pedrycz and Gomide, 1998; Kosko, 1992; Klir and Folger, 1988; Kandel, 1982; Dubois and Prade, 1980). Examples of such scalar descriptors include height, support, and core. Consider fuzzy set A shown in Figure 2 where all these notions are visualized.

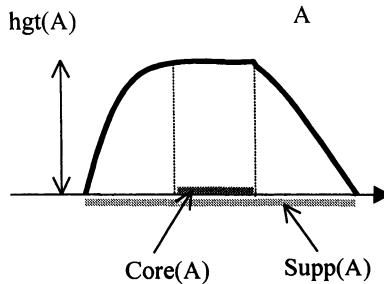


Figure 2. Fuzzy set A and its height, core and support.

By the height of A , $\text{hgt}(A)$ we mean a supremum of the membership function (in case of the finite universe of discourse X we take a maximal value of its membership function). The element of X for which this supremum occurs can be treated as the most typical for the fuzzy set (we say that this element is a prototype of A). If $\text{hgt}(A)$ is equal to 1 we consider A to be a normal fuzzy set. If this does not hold, we call it subnormal. In the sequel, we show that the subnormality property is usually a result of dealing with concept for which we cannot identify any element that fully satisfies it. This happens quite often in decision-making where the resulting fuzzy set of alternatives does not include any ideal element that fully meets our objectives (where goals and constraints are usually in conflict).

A *support* of A (denoted by $\text{Supp}(A)$) is a collection of all elements of X belonging to A with a nonzero membership degree. Formally speaking we have

$$\text{Supp}(A) = \{x \in X \mid A(x) > 0\}$$

By the *core* of A (denoted by $\text{Core}(A)$) we mean all elements of X for which $A(x)$ is equal to 1, that is

$$\text{Core}(A) = \{x \in X \mid A(x) = 1\}$$

Note that these two descriptors of A return a set rather than a fuzzy set. An important and general notion that relates fuzzy sets to sets is the one of an α -cut. An α -cut of A , denoted by A_α , is a set which results from “cutting” A at the given level α ($\alpha \in [0,1]$). We have the following definition

$$A_\alpha = \{x \in X \mid A(x) \geq \alpha\}$$

Assuming that the threshold level α describes our preference as to the “meaningful” (that is strong enough) membership to the fuzzy set, it is obvious that A_α captures the elements whose membership to the fuzzy set is beyond discussion.

As α can take on any value in $[0,1]$, for each fuzzy set we have an infinite family of α -cuts. It means that a single fuzzy set induces a family of sets indexed by a certain value of α . The obvious inclusion holds: if α increases, this implies smaller, more confined sets. If $\alpha=1$ we get the core of A . Computing α -cuts of A leads to the conversion of A to a certain set. By choosing high values of the threshold (viz. α), we take into consideration the most significant portion of the fuzzy set.

There is a fundamental relationship between any fuzzy set and a family of sets that is a representation theorem (Klir and Folger, 1988; Zimmermann, 2001). This theorem states that A is *represented* by an infinite family of its α -cuts that is

$$A = \bigcup_{\alpha \in [0,1]} \alpha A_\alpha \quad (2)$$

Rewriting the above expression in terms of the membership functions we have the following relationship

$$A(x) = \sup_{\alpha \in [0,1]} [\alpha A_\alpha]$$

This theorem formalizes a fact that is intuitively appealing: we require a family of sets to represent a fuzzy set and this relationship points at the generality of fuzzy sets. Sets are special cases of fuzzy sets. Fuzzy sets subsume sets. One can look at

the representation theorem from a different and more conceptual standpoint. Fuzzy sets generalize sets and this extension can be quantified in the form of the cutworthy property that is an essence bridges fuzzy sets with set theory (Klir, 2001).

Bringing the geometry of fuzzy sets (Kosko, 1992) into investigation makes this point even more profound. To come up with a more amenable interpretation, we confine ourselves to finite universe of discourse, $\text{card}(X) = n$. This helps us treat any fuzzy set as a point in the n -dimensional hypercube, $A \in [0,1]^n$. In this geometry, sets are vertices of the unit hypercube $\{0,1\}^n$. An illustration for $n = 2$ is shown in Figure 3. Note that the four sets are the corners of the unit square. On the other hand, fuzzy sets spread across the entire square as indicated in the form of the shadowed interior of the square.

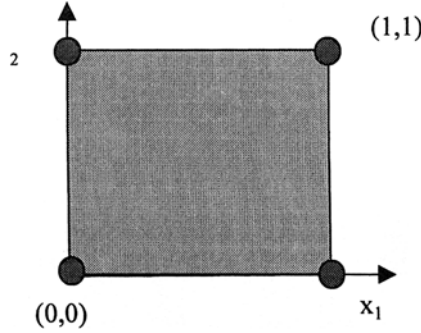


Figure 3. Geometric representation of sets and fuzzy sets for $n = 2$; in particular $(0,0)$ denotes an empty set while $(1,1)$ describes the entire universe.

Some further descriptors of fuzzy sets concern the shape of their membership functions; the three of them are commonly used

A is a *unimodal* fuzzy set if its membership function exhibits a single maximum. In other words, there exists only a single point x_0 where $\text{hgt}(A)$ is attained.

A is *convex* if its membership function satisfies the relationship

$$A(\lambda x_1 + (1-\lambda)x_2) \geq \min(A(x_1), A(x_2)) \quad (3)$$

for any x_1, x_2 in X where $\lambda \in [0,1]$.

Analogously we introduce the notion of concavity.

Fuzzy set A is *concave* if its membership function satisfies the condition

$$A(\lambda x_1 + (1-\lambda)x_2) \leq \min(A(x_1), A(x_2))$$

for any x_1, x_2 in X where $\lambda \in [0,1]$.

3.3 MAIN CLASSES OF MEMBERSHIP FUNCTIONS

As we stated earlier, the way in which partial membership is represented depends on the problem and a suitable membership function needs to be selected according to the application at hand. We have already discussed a few examples, refer to the previous section. Here we show detailed expressions for the membership functions

Triangular fuzzy sets

$$A(x; a, m, b) = \begin{cases} 0 & \text{if } x \leq a \\ \frac{x-a}{m-a} & \text{if } x \in [a, m] \\ 1 - \frac{b-x}{b-m} & \text{if } x \in [m, b] \\ 0 & \text{if } x \geq b \end{cases} \quad (4)$$

In the above class of fuzzy sets, a, m , and b are the parameters describing the linear segments of this membership function. We can rewrite the membership in the concise format using the operations of minimum and maximum

$$A(x; a, m, b) = \max \{ \min [(x-a)/(m-a), (b-x)/(b-m)], 0 \}$$

Trapezoidal fuzzy sets

$$A(x; a, m, n, b) = \begin{cases} 0 & \text{if } x \leq a \\ \frac{x-a}{m-a} & \text{if } x \in [a, m] \\ 1 & \text{if } x \in [m, n] \\ 1 - \frac{b-x}{b-m} & \text{if } x \in [n, b] \\ 0 & \text{if } x \geq b \end{cases} \quad (5)$$

Gaussian fuzzy sets

$$A(x; m, \sigma) = \exp(-(x-m)^2/\sigma^2)$$

Nonsymmetric Gaussian fuzzy sets

$$A(x; m, \sigma, \mu) = \begin{cases} \exp(-(x-m)^2/\sigma^2) & \text{if } x \leq m \\ \exp(-(x-m)^2/\mu^2) & \text{if } x > m \end{cases} \quad (6)$$

Parabolic fuzzy sets

$$A(x, m, p) = \begin{cases} 1 - p^2(x - m)^2 & \text{if } x \in [m - 1/p, m + 1/p] \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

The plots of these membership functions are shown in Figure 4; it becomes evident that with fuzzy sets we are provided with panoply of expressing a form in which a transition between full membership and full exclusion occurs.

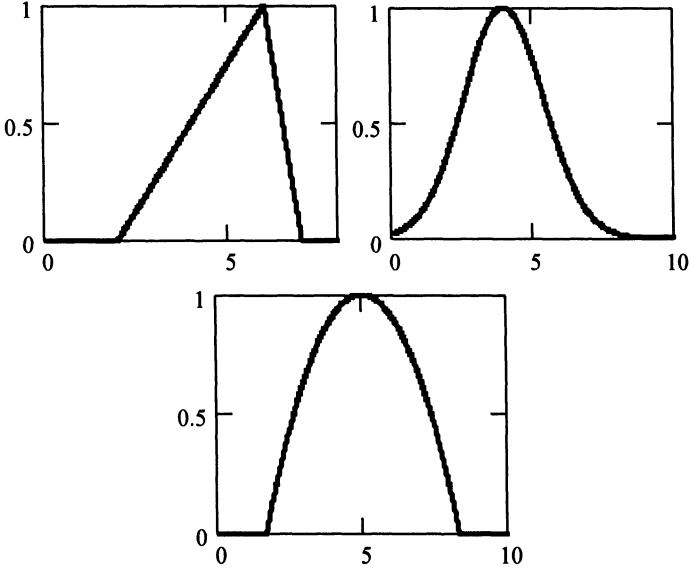


Figure 4. Examples of typical classes of fuzzy sets: triangular, Gaussian,, and parabolic.

A certain type of fuzzy set can be chosen quite quickly realizing the essence of the transition captured by each membership function. Apparently, we usually have a preliminary idea on how the nature of expressions like *around*, *small*, *large*, etc. could be captured. On a practical note, one should think of possible implications of this selection. Simplicity is usually a prudent criterion worth discussing. One has to consider how much domain knowledge will be available and at which level. This usually is a dominant factor when choosing between triangular (trapezoidal) membership functions and some more sophisticated membership functions. Consider a fuzzy set realizing a concept of acceptable delay being used in some manufacturing process. If we know that 10 hours is our upper limit of this delay and at the same time we do not have any specific knowledge as to the level of acceptability within this range then a simple linear model is a feasible option, see Figure 5 (a). On the other, hand if we anticipate that our acceptability level could drop dramatically when moving close to the upper limit of 10 hours, then a nonlinear

membership function emphasizing this effect could be a good option, see Figure 5 (b). As a result we may end up with a nonlinear membership function.

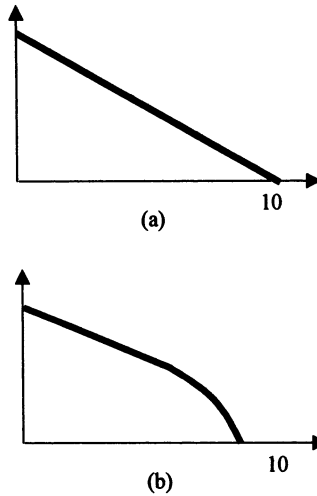


Figure 5. Selecting a class of membership functions representing a concept of acceptable delay: linear membership function (a) and a nonlinear membership function emphasizing a rapid drop of acceptability level at values close to 10 (b).

The choice of the class of the membership functions depends also on the experimental data available (we will discuss the matter in terms of membership function estimation). Then choosing a membership function with some adjustable parameters is worth considering. In this way we can adjust the values of these parameters so that the membership function fits the experimental data.

There is another design aspect of fuzzy sets that is a sensitivity of the membership functions. It is computed in a usual way as the absolute values of the derivative of

the membership function $\left(\left| \frac{\partial A}{\partial x} \right| \right)$ and expresses how the given membership function

are affected. These changes can be attributed to the distribution of the membership grades when moving along "x". There are different distributions of the sensitivity function as shown in Figure 6. Triangular membership functions come with a constant sensitivity for all elements located at the same side of the modal value that is equal to the slope of the membership function in this particular region. The Gaussian membership function has the distribution of sensitivity that is more diversified. It is equal to zero at the modal value of the membership function (that is a desirable effect considering that the modal value should be treated as the representative of the fuzzy set). The maximal level of sensitivity is attained for the membership grade being equal to 0.5. This pattern of behavior implies that the

membership grades around this value are quite susceptible to changes and errors caused by the use of some estimation procedure. The parabolic membership function exhibits an interesting sensitivity pattern that is we encounter a linear change in the sensitivity values with the lowest value located with the high membership degrees; the sensitivity achieves zero for the membership grade equal to one.

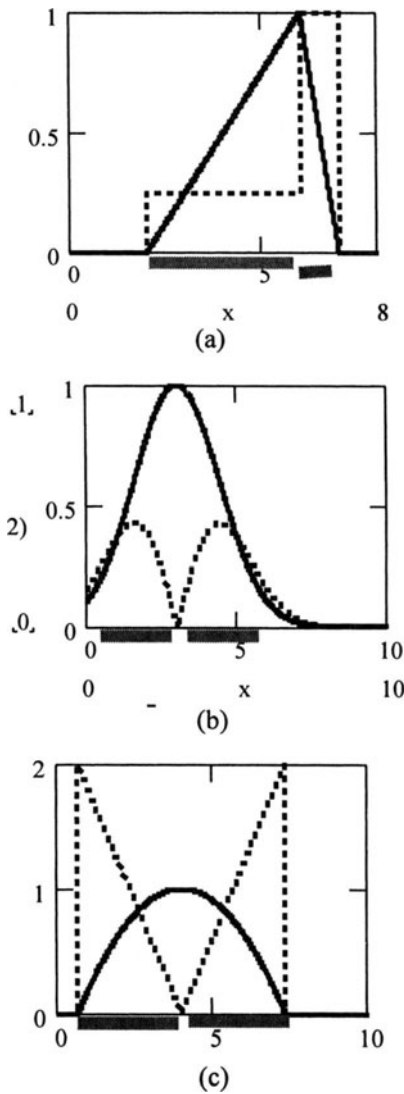


Figure 6. Sensitivity of several classes of membership functions: triangular (a), Gaussian (b), and parabolic (c). Solid line- membership function; dotted line – sensitivity function. Identified are several regions of sensitivity of the fuzzy sets.

Summarizing, in the selection of the membership functions, we are guided by the following criteria

- available domain knowledge
- simplicity of the membership function
- possible parametric optimization of the fuzzy sets (calibration of the membership function)

In this setting, it also worth stressing that fuzzy sets are highly context dependent and this property calls for the calibration of their membership functions. The calibration (adjustment) takes place both at the horizontal and vertical level that is it involves the modification (transformation) of the universe of discourse in which they are defined as well as the membership grades. The fact of context dependency is an evident asset of fuzzy sets; the same concept can be used across various environments. Say, *low* inflation is useful in describing economies of various countries and has the same semantics but needs to be calibrated (eventually by changing the universe of discourse).

3. 4 OPERATIONS ON FUZZY SETS

For fuzzy sets defined in the same space we define standard operations of union, intersection, and complement. Historically, the first operations on fuzzy sets were defined using the same minimum and maximum operation we used for characteristic functions in case of set theory. This means

$$A \cap B \quad (A \cap B)(x) = \min(A(x), B(x)) \quad (8)$$

$$A \cup B \quad (A \cup B)(x) = \max(A(x), B(x)) \quad (9)$$

The complement of A (denoted by \overline{A}) is standard and comes in the form $1-A(x)$.

As the membership values are in the unit interval, the way in which logic operations are developed is far beyond the two operations shown above. As a matter of fact, this phenomenon has been identified quite early in the development of fuzzy sets. For instance an intersection was modeled by a product operation meaning that the membership function of the intersection of A and B is realized as $A(x)B(x)$. A general class of logic connectives in fuzzy sets comes in the form of triangular norms (t-norms and s-norms). These originally coming from the theory of probabilistic metric spaces (Schweizer and Sklar, 1983) were adopted in fuzzy sets as a sound models of logic connectives. The minimal set of properties to be met by any models of operators of logic connectives consists of commutativity,

associativity, monotonicity and some boundary conditions. Let us elaborate on them in more detail

The formal definition of t- and s-norm is as follows (Klement et al., 2000; Butnariu and Klement, 1993)

A t-norm is a two-argument function $t: [0,1] \times [0,1] \rightarrow [0,1]$ that satisfies the following conditions

$a \ t \ b = b \ t \ a$	commutativity
$a \ t \ (b \ t \ c) = (a \ t \ b) \ t \ c$	associativity
if $a < a'$ and $b < b'$ then $a \ t \ b < a' \ t \ b'$	monotonicity
$a \ t \ 0 = 0 \quad a \ t \ 1 = a$	boundary condition

$a, b, c, a', b' \in [0,1]$

We realize that the properties of the t-norm correspond with the list of the formal requirements expressed for any possible realization of the *and* operation (intersection)

The definition of an s-norm involves the same conditions as for the t-norms with an exception of the boundary conditions that need to be revised. Formally we have

An s-norm is a two-argument function $s: [0,1] \times [0,1] \rightarrow [0,1]$ that satisfies the following set of conditions

$a \ s \ b = b \ s \ a$	commutativity
$a \ s \ (b \ s \ c) = (a \ s \ b) \ s \ c$	associativity
if $a < a'$ and $b < b'$ then $a \ s \ b < a' \ s \ b'$	monotonicity
$a \ s \ 0 = a \quad a \ s \ 1 = 1$	boundary condition

$a, b, c, a', b' \in [0,1]$

A selected collection of t- and s-norms is summarized in Table 1. The two first are in common use; in particular the min and max operators are well known and easy when it comes to their digital realization (require only comparisons and selection operations). The third one for $p=0$ produces well known Lukasiewicz logic connectives being used in his seminal work on multivalued logic. In this case we have $x \ y = \max(0, x+y-1)$ and $x \ s \ y = \min(1, x+y)$. The last pair of the t- and s-norms in Table xx is known as a drastic product and drastic sum, respectively.

Different t- and s-norms are realizations of the same logic operation (*and* and *or*) on two fuzzy sets. The rationale behind having a collection of realizations rather than having a single realization (say min and max, respectively) is to gain more flexibility to reflect the semantics of operators as being used in different contexts. To support this line of thought, some of the t-norms are parameterized and this contributes to their flexibility when it comes to modeling logic connectives for experimental data.

To quantify this flexibility, the plots of the third t-norm treated as a function of its parameter (p) for selected combinations of the arguments ($x=x_0$ and $y=y_0$) are shown in Figure 7. Evidently, we note substantial changes in the results of this type of logic aggregation that are controlled through a suitable selection of the parameter; the higher values of the parameter lead to the lower values of the aggregation. Furthermore when the values of “ p ” are above 10, they do not affect the result of the t-norm.

t-norm	s-norm
$\min(x, y)$	$\max(x, y)$
xy	$x + y - xy$
$\max(0, (1+p)(x+y-1)-pxy), p \geq -1$	$\min(1, x+y+pxy), p \geq -1$
$1 - \min(1, \sqrt[p]{(1-x)^p + (1-y)^p}), p \geq 0$	$\min(1, \sqrt[p]{x^p + y^p}), p \geq 0$
$\frac{xy}{p + (1-p)(x+y-xy)}, p \geq 0$	$\frac{x+y-xy-(1-p)xy}{1-(1-p)xy}, p \geq 0$
$\frac{1}{\sqrt[p]{\frac{1}{x^p} + \frac{1}{y^p} - 1}}, p > 0$	$1 - \frac{1}{\sqrt[p]{\frac{1}{(1-x)^p} + \frac{1}{(1-y)^p} - 1}}, p > 0$
$\frac{xy}{\max(xy, p)}, p \in [0, 1]$	$1 - \frac{(1-x)(1-y)}{\max((1-x)(1-y), p)}, p \in [0, 1]$
$\begin{cases} x & \text{if } y = 1 \\ y & \text{if } x = 1 \\ 0, & \text{otherwise} \end{cases}$	$\begin{cases} x & \text{if } y = 0 \\ y & \text{if } x = 0 \\ 1, & \text{otherwise} \end{cases}$

Table 1. A list of selected t- and s-norms

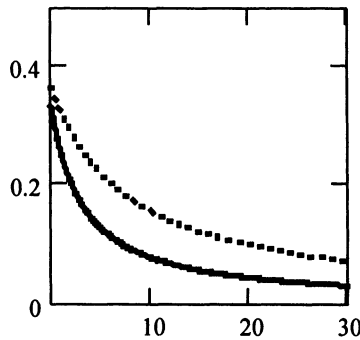


Figure 7. Plots of the t-norm (third row) as a function of “ p ” for two selected combinations of values of x_0 and y_0 : $x_0=0.5, y_0=0.5$ (solid line) and : $x_0=0.8, y_0=0.4$ (dotted line).

One of the interesting features distinguishing between different t-norms is their interactivity or compensation effect. Descriptively, we consider an operation to be interactive if the result of it depends on the values of the membership grades involved. Some t-norms are more interactive than others. The interactivity occurs when combining fuzzy sets. For instance, let X be a universe of discourse of different car makes. A is a fuzzy set of expensive vehicles B is a fuzzy set of reliable vehicles. The compound statement “*expensive and reliable vehicles*” translates into an intersection of A and B . Intuitively, we sense that A and B interact quite vigorously in the sense that these two features (reliability and cost) are strongly associated (viz. they strongly interact). Now consider C to be a fuzzy set of *new* cars and D formed as *fast* cars. An intersection of C and D results in a fuzzy set where there is no evident interaction (or the interaction that is so strong as in the previous case). As a simple example, consider a process of selecting a car make on a basis of two criteria such as reliability and price. Intuitively, we can envision that these two criteria are somewhat in conflict; it is likely that low price (that we would like to have) may not buy us high reliability. While this becomes evident, fuzzy sets help quantify the effect of conflict and become instrumental in identifying possible decisions. The two fuzzy sets (defined in the finite universe of discourse of car makes) are shown below

car make	x_1	x_2	x_3	x_4	x_5
price (A)	1.0	0.7	0.4	0.5	0.8
reliability(B)	0.3	0.8	1.0	0.1	0.2

As we are interested in the satisfaction of the two criteria, we take the intersection of A and B . The results are shown for the minimum and product

car make	x_1	x_2	x_3	x_4	x_5
A and B (min)	0.3	0.7	0.4	0.1	0.2
A and B (product)	0.3	0.56	0.4	0.05	0.16

While the minimum shows a lack of compensation, both operations point at the same car make (shadowed entry of the table) preferred in light of the assumed criteria; it is also important to note that the height of the intersection is 0.56 or 0.7 that is quite substantial; it tells us that the selected make meets quite well both criteria.

For each t-norm we have a dual s-norm that is defined as follows

$$a \text{ s } b = 1 - (1-a) \text{ t } (1-b) \quad (10)$$

$a, b \in [0, 1]$. And reciprocally, we have $a \wedge b = 1 - (1-a) \vee (1-b)$. These are just the well-known DeMorgan laws.

With the t - and s -norm realizations of the intersection and union on fuzzy sets, we can revisit the properties of these operations. All of the properties we have for sets (refer to the discussion in Chapter 2) hold here as well with an important exception

$$A \cap \bar{A} \neq \emptyset$$

and

$$A \cup \bar{A} \neq X$$

These two are referred to as an overlap and underlap property as they imply the following inequalities $A \cap \bar{A} \supseteq \emptyset$ and $A \cup \bar{A} \subseteq X$. They are inherently implied by the intermediate membership grades different from 0 and 1. In a nutshell, (xx) states that the Aristotelian principle (yes-no dichotomization) does not hold for fuzzy sets. In an extreme situation when A has a constant membership function equal to $1/2$, we have $(A \cup \bar{A})(x) = 1/2$ and $(A \cap \bar{A})(x) = 1/2$.

3. 5 INFORMATION GRANULARITY AND FUZZY SETS

Fuzzy sets can be conveniently characterized by two scalar indexes describing their size (granularity) and uncertainty (viewed from the standpoint of hesitation of assigning elements to the fuzzy set). The first indicator is very much in line with the cardinality of sets introduced in Chapter 2 while the second one relates to the notion of entropy as a measure of choice (selection). We usually refer to these two indicators (or classes of indicators) as energy and entropy measures of fuzziness. More formally, an energy measure of fuzziness of A defined in X and denoted by $E(A)$ is defined as the following expression (Capocelli and Luca, 1973; Ebanks, 1983; Knopfmacher, 1975; Gottwald, 1979)

$$E(A) = \int_X e(A(x)) dx \quad (11)$$

where $e: [0,1] \rightarrow [0, 1]$ is an increasing function. (in case of final space X , the above integral is replaced by the summation operation). In essence, this operation index is about counting elements in the fuzzy set. While counting elements in any set does not raise hesitation, carrying out the same counting process for fuzzy sets requires some rules on how to assess to which extent each element belongs to the formation of the energy of the fuzzy set. The only observation we can agree upon is that higher membership grades should gain more "visibility" and this effect is modeled through different types of functions (e). In the simplest case we consider this to be an identity function, $e(u) = u$ and this raises to the notion of cardinality (a

so-called σ -count of a fuzzy set) of A where all elements (membership grades) are added. More general forms of the aggregation functions (e) include such monotonic operations as $e(u)=u^p$ with $p > 0$ or $e(u) = \sin(\pi u/2)$. By using these type of relationships, we affect the way in which specific membership values contribute to the energy of fuzziness, say u^p with $p=2$ discounts all membership grades while an opposite effect occurs for $p<1$. From this analysis we conclude that the energy measure of fuzziness is a direct instrument that helps us measure the granularity of the information granule (fuzzy set). The higher the information granule, the lower the values of the energy measure of fuzziness. Obviously, a detailed quantification of this relationship depends upon the form of the aggregation function and the type of the membership function itself. Some examples of this type of relationship are shown in Figure 8 where we consider a Gaussian membership function with different spreads and choose several cases of the aggregation function $e(u)=u^p$ with $p=0.5, 1$, and 2 .

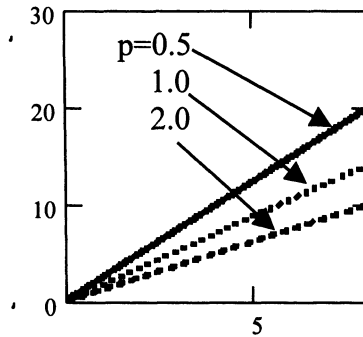


Figure 8. Energy measure of fuzziness of the Gaussian fuzzy set for selected values of the exponent p .

The entropy measure of fuzziness $H(\cdot)$ (Trillas and Riera, 1978) is associated with a way of expressing hesitation as to the membership grades of the individual elements belonging to the fuzzy set. The structure of this measure is the same as for the energy measure of fuzziness. We have

$$H(A) = \int_x h(A(x)) dx \quad (12)$$

where $h(u): [0,1] \rightarrow [0,1]$ is a continuous function satisfying the property of symmetry $h(u)=h(1-u)$ and monotonicity in the sense $h(u)$ is monotonically increasing over $[0,1/2]$ and monotonically decreasing over $[1/2, 1]$ with the boundary conditions $h(0)=h(1)=0$ and $h(1/2)=1$. A closer inspection of the properties of the aggregation function (h) reveals that we strongly value the membership grades that are the most

“uncertain” as being equal to $\frac{1}{2}$. This is intuitively appealing as this membership grade raises the highest hesitation as to the accepting or rejecting this element as belonging to A. Note that there is no hesitation when it comes to elements with full belongingness ($h(u)=1$) or full exclusion ($h(u)=0$). As before, a choice of a certain type of the aggregation function helps express a way in which the uncertainty is taken into consideration in the overall expression (12). The commonly encountered forms of $h(\cdot)$ are

$$h(u) = -u \log(u) - (1-u) \log(1-u)$$

(this form of “h” is the same as being used in the calculations of entropy in the probability theory)

$$h(u) = 4u(1-u)$$

$$h(u) = \begin{cases} 2u & \text{if } u \in [0, 1/2] \\ 2(1-u) & \text{if } u \in [1/2, 1] \end{cases}$$

Owing to the symmetry of “h” we immediately have $H(A) = H(\bar{A})$ meaning that A and its complement have the same value of the entropy function.

Some illustrative plots of the entropy measure of fuzziness for the Gaussian fuzzy sets and the aggregation function defined as $4u(1-u)$ are shown in Figure 9.

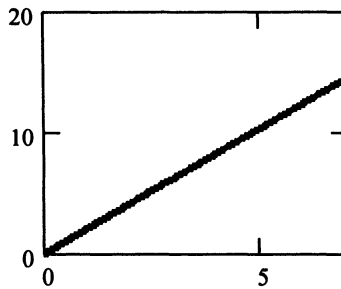


Figure 9. Entropy of the Gaussian fuzzy set with $m=2$ treated as a function of σ .

There is an interesting measure that relates to the problem of quantifying hesitation of selecting a single element from the fuzzy set to be treated as its representative. Intuitively, if A consists of a single element, there is no hesitation to choose this element as a representative of A. On the other hand, if A covers X (viz. its membership function is equal identically to 1 over this space) then we are faced with the highest level of hesitation. To measure this phenomenon, in (Yager, 1983) proposed was a measure of specificity, $Sp(\cdot)$ that maps A into a single number $Sp(A)$

such that (a) $Sp(A) = 1$ if and only if there exists only one element of X where $A(x) = 1$ and zero otherwise, (b) if $A \subset B$ then $Sp(A) \geq Sp(B)$. The integral of the following form is viewed as a realization of this measure

$$Sp(A) = \int_0^{\alpha_{\max}} \frac{d\alpha}{Card(A_\alpha)} \quad (13)$$

where α_{\max} is the height of A .

3. 6 RELATIONSHIPS BETWEEN FUZZY SETS IN THE SAME SPACE

Having two fuzzy sets defined in the same space, we can characterize a relationship between them. There are two fundamental notions, that is possibility and necessity measures (Zadeh, 1978; Dubois and Prade 1980). Let X and A be two fuzzy sets in X . The possibility measure, $Poss(X,A)$ is computed as

$$Poss(X,A) = \sup_{x \in X} [\min(X(x), A(x))] \quad (14)$$

This measure expresses an extent to which two fuzzy sets overlap. It can be then viewed as a certain measure of matching between X and A . The measure is symmetric, $Poss(X,A) = Poss(A, X)$ and monotonic with respect to its two arguments, $Poss(X,A) \leq Poss(X', A)$ when $X \subset X'$.

The necessity measure, $Nec(X,A)$ is defined as

$$Nec(X,A) = \inf_{x \in X} [\max(1-X(x), A(x))] \quad (15)$$

Referring to Figure 10, we note that this measure expresses an extent to which X is included in A so it can be viewed as a measure of containment for two fuzzy sets.

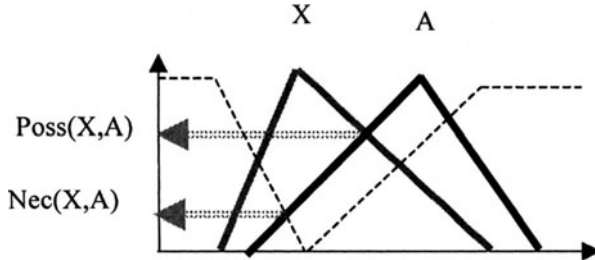


Figure 10. Possibility and necessity measures computed for two fuzzy sets A and X ; dotted line shows a complement of X used in the computations of the necessity measure.

In contrast to the possibility measure, the necessity measure is asymmetric, that is $\text{Nec}(X,A) \neq \text{Nec}(A,X)$.

There is an interesting inverse problem related to these two measures that pertains to the reconstruction task. It is formulated as follows: fuzzy set A is given along with the possibility and necessity measure, $\lambda = \text{Poss}(X, A)$ and $\mu = \text{Nec}(X,A)$. Reconstruct the second fuzzy set (X). We show that this reconstruction plays an important role in communication problems between granular words. Some solutions to it can be easily expressed in the language of fuzzy relational equations (Di Nola et al., 1989).

3. 7 FUZZY SETS AND LINGUISTIC VARIABLES

Fuzzy sets come with a clearly defined semantics as operational entities quantifying linguistic terms. In this way, we are provided with a formal framework to work with information granules. One can move this point of view even further not considering a single fuzzy set but looking at a family of fuzzy sets defined in the same space. In this case, we are concerned with a linguistic variable that is a variable that assumes granular values (that is fuzzy sets). The formal structure arising in this manner (Zadeh, 1975) comes in the form

$$\langle \text{variable_name}, T, X, \mathcal{G}, \mathcal{M} \rangle \quad (16)$$

where *variable_name* is a name of the variable (say temperature, pressure, inflation, speed, etc), T denotes a finite term set (that is a collection of labels – names being of interest when this variable is being used) and X is a universe of discourse over which the variable is defined. \mathcal{G} is syntax of the variable that expresses how new terms can be generated on a basis of the existing term set and some logic operators; usually the syntax is defined in the form of some grammar. Finally \mathcal{M} is concerned with the semantics of the term and in essence it shows rules of determining membership functions of the new terms on a basis of those already available and the meaning of the operators on fuzzy sets. It is worth stressing that the family of the generic terms is usually limited to 7 ± 2 (as motivated by the findings in Miller (Miller, 1956)). An illustrative example of the linguistic variable of temperature is shown in Figure 11; we show different components of the structure introduced above are used.

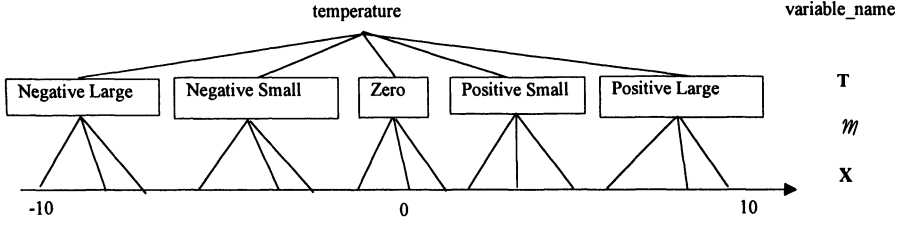


Figure 11. A structure of a linguistic variable of temperature.

In the context of linguistic variables, there are two important operations of linguistic modification known as modifiers or hedges. They correspond to the linguistic terms of *very* and *more or less*, say *very low temperature*. The semantics of these modifiers is captured as the following transformations known as a concentration and dilation

$$(\text{very } (A))(x) = A^2(x)$$

$$(\text{more or less } A)(x) = A^{0.5}(x)$$

where these operations affect the membership functions by concentrating its values (the first case) or raising the values of the membership function (dilation). In general, one can look at the power transformation A^r where $r \geq 0$ and all values of r less than 1 imply a certain dilation effect while $r > 1$ causes some concentration effect.

3. 8 TRANSFORMATIONS OF FUZZY SETS BETWEEN SPACES

So far we discussed operations on fuzzy sets defined in the same space. To make fuzzy sets operational, we need to describe how the fuzzy sets are transformed between spaces. This generalizes the well-known concept of any numeric mapping between two spaces, say $f: X \rightarrow Y$ that is $y = f(x)$. Now if the elements in this mapping are fuzzy sets rather than numbers, a question arises as to the mapping $f(A)$. Intuitively, when transforming A through some function, we anticipate the result to be another fuzzy set occurring in Y . The transformation of fuzzy sets realized through “ f ” is known as an extension principle. The membership function of B where $B=f(A)$ is computed as

$$B(y) = \sup_{x \in X: y=f(x)} [A(x)] \quad (17)$$

An example of the calculations of the membership function is illustrated in Figure 12. For any fixed membership value $B(y)$ we first identify all x 's for which $y=f(x)$ (which in essence is an inverse problem). Say these are $x(1)$, $x(2)$, $x(3)$ and $x(4)$. For each of them we compute the membership grade that is $A(x(1))$, $A(x(2))$... and take

the maximum out of them and assign this value to be the membership grade of $B(y)$. In the case shown in Figure 12 this $x(2)$ so $B(y) = A(x(2))$.

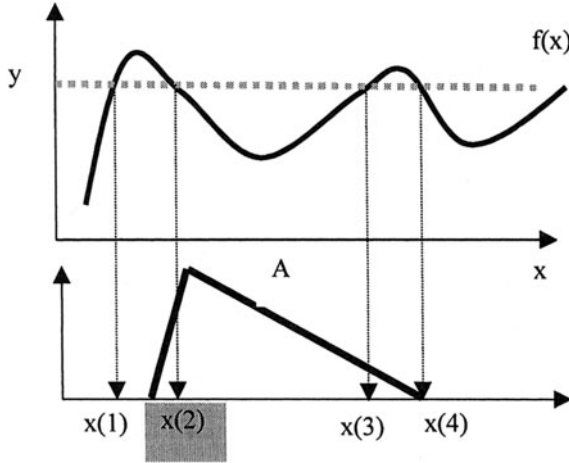


Figure 12. An illustration of the extension principle; note that $x(2)$ comes with the maximal value of $A(x)$ and this is treated as the membership function of $B(y)$.

It is worth underlining the essence of the transformation of the fuzzy sets. The essence of (17) is a nonlinear optimization program in which we find the supremum of the expression $A(x)$ under some nonlinear constraint expressed by the underlying function “ f ” that is

$$\begin{aligned} B(y) &= \sup_{x \in X} [A(x)] \\ \text{subject to} \\ y &= f(x) \end{aligned}$$

The use of the supremum (or maximum) operation is to assure the uniqueness of the solution in case we encounter more than a single solution to the equation $y = f(x)$ for a fixed value of “ y ”

The extension principle extends to multivariable functions, say $y = f(x_1, x_2, \dots, x_n)$ where A_1, A_2, \dots, A_n are fuzzy sets defined in the corresponding input spaces X_1, X_2, \dots, X_n . We have $B = f(A_1, A_2, \dots, A_n)$ whose membership function is computed as

$$\begin{aligned} B(y) &= \sup_{\substack{x_1 \in X_1 \\ x_2 \in X_2 \\ \dots \\ y = f(x_1, x_2, \dots, x_n)}} [\min(A_1(x_1), A_2(x_2), \dots, A_n(x_n))] \end{aligned}$$

One can view this as a nonlinear optimization problem that arises in the following format

$$\begin{aligned}
 B(y) = \sup_{\substack{x_1 \in X_1 \\ x_2 \in X_2 \\ \dots}} [\min(A_1(x_1), A_2(x_2), \dots, A_n(x_n))] \\
 \text{subject to} \\
 y = f(x_1, x_2, \dots, x_n)
 \end{aligned}$$

with the constraint being implied by the multivariable function “f”. There are also some generalizations of the extension principle along the line of using any t-norm instead of the minimum operations. This induces a certain interactivity effect however it comes with a certain computing overhead.

The role of the extension is central to processing fuzzy sets and computing a way in which they propagate across functions. We show its direct use when dealing with fuzzy numbers and fuzzy arithmetic.

3.9 FUZZY ARITHMETIC

Fuzzy arithmetic is concerned with processing fuzzy sets defined in the reals (\mathbf{R}) whose underlying transformations are just algebraic operations (addition, subtraction, multiplication and division). Let us recall that fuzzy numbers are granular generalizations of real numbers and intervals. In this sense the ensuing discussion closely relates to the interval analysis discussed in the previous chapter. Fuzzy numbers are unimodal fuzzy sets defined in \mathbf{R} with continuous membership functions. Usually depending on specific application we impose some additional requirements such as continuity of the membership functions of the fuzzy numbers. Examples of fuzzy numbers are shown in Figure 13. Triangular fuzzy numbers are convenient to deal with and carry enough expressive power. As shown in Figure 13 (a) they are fully described by a triple (a, m, b) with “m” denoting a modal value and “a” and “b” being the bounds of the support.

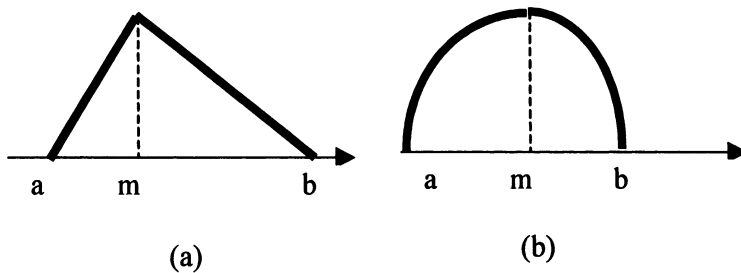


Figure 13. Example membership functions of fuzzy numbers.

We are interested in the algebra of fuzzy numbers. Let us start with addition of A and B with the two being triangular fuzzy numbers. The result, denoted as $C = A \oplus B$ (where \oplus is used to denote that we are concerned with the operation on fuzzy numbers not real numbers). The membership function of C results from the extension principle. We have

$$C(z) = \sup_{x,y \in \mathbb{R}: z=x+y} [\min(A(x), B(y))] = \sup_x [\min(A(x), B(z-x))]$$

As we are dealing with triangular membership functions A and B, these calculations can be made more specific. It is instructive to complete the calculations for the parameters of the resulting membership function (that appears to be a triangular membership function as well). In the calculations, we will separately treat the increasing and decreasing slopes of the membership functions of the two fuzzy numbers; the details are shown in Figure 14. Consider a certain membership grade of the result (C) and denote it be w. This means that there exists some value of “x” and “y” for which the following relationships hold

$$w = \frac{x-a}{m-a} \quad . \quad w = \frac{y-c}{n-c}$$

As $z=x+y$ we have

$$z = w(m-a)+a +w(n-c)+c$$

From this we determine the value of “w” treating this as a function of “z” (that is nothing but an expression of the membership function of the rising segment of C). Rearranging the variables we get

$$w = \frac{z - (a+c)}{m+n - (a+c)}$$

which is nothing but a linear expression of z. In other words, the increasing part of the membership function of C defined over the range $[a+c, m+n]$ is a linear function of “z” with two parameters (a+c) as the lower bound and (m+n) as the modal value of this result.

Proceeding with the decreasing parts of the membership functions of A and B we arrive at the formula

$$w = 1 - \frac{z - (m+n)}{(b+d) - (m+n)}$$

which is again a linear function of “z”. We recognize that the modal value is equal to $m+n$ with the upper bound given as $b+d$. Using a concise notation, we conclude that the result of addition of two triangular fuzzy numbers (and any finite number of such fuzzy numbers) is another triangular fuzzy number whose parameters express via the sum of the corresponding parameters of the contributing fuzzy sets),

$$A(a,m,b) \oplus B(c,n,d) = C(a+c, m+n, b+d) \quad (18)$$

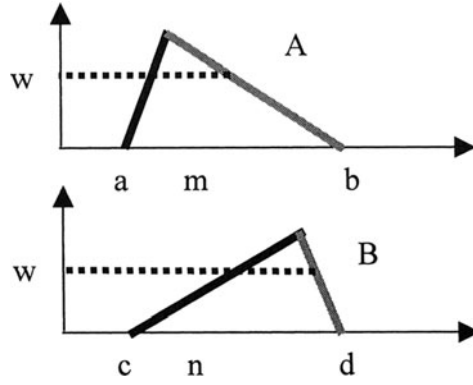


Figure 14. Detailed computations of the membership function of the sum of two fuzzy numbers.

The calculations for the remaining algebraic operations are carried out in an analogous manner. It is worth stressing that the resulting fuzzy number is not described by a triangular membership function.

3. 10 FUZZY RELATIONS AND RELATIONAL CALCULUS

Relations are fundamental notions of mathematics, science and engineering. Observations of the form “a and b are related” are in the center of system modeling, analysis and design. Fuzzy relations, sometimes referred to as multidimensional fuzzy sets admit notions with partial dependency. For instance, the statements “a and b are approximately equal”, “John is taller than Jim”, “car-1 is faster than car-2”, etc describe some sort of relationship (dependency) between objects. Moreover we sense that some instances may satisfy partially such relationships. Formally, we can think of any fuzzy relation as a descriptor of the statement “a and b are related” where the term “related” can be instantiated as one of the examples shown above. The membership function of a relation R shows how objects are related. Formally, a fuzzy relation R defined in the Cartesian product X and Y, that is described by a membership function

$$R: X \times Y \rightarrow [0,1]$$

where $R(x,y)$ indicates an extent to which x and y are related. The relation “approximately equal” can be described in the form $R(x,y) = \exp(-(x-y)^2/\sigma^2)$ where σ controls a spread of the membership function. As shown in Figure 15, for $x=y$ we get the membership degree equal to 1. In contrast the Boolean relation “equal to” is also included in the same figure.

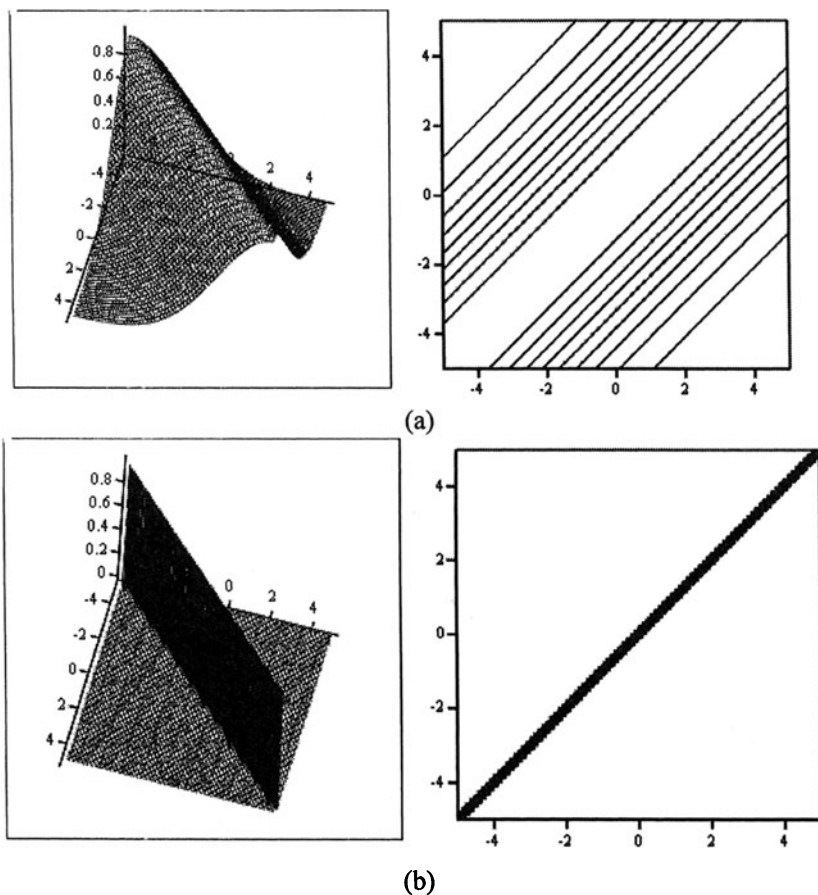


Figure 15. Membership function (3D and contour plots) of the fuzzy relation “approximately equal” and its Boolean counterpart of the relation “equal to”.

The notion of relation extends to any number of arguments and then we end up with multivariable membership functions, say $R(x,y,z,w)$, etc.

Depending on the semantics, fuzzy relations are direction-free constructs and this corresponds to their symmetry. This means that there is no specific “direction” as to the causal dependency between its arguments. For instance, the relation “approximately equal” is symmetric and does not exhibit any directionality. On the other hand, functions exhibit an evident directionality. By stating $y = f(x)$ we mean that x implies y so y is a dependent variable. We may say that functions capture the notion of causality where we express that occurrence of some value of “ x ” causes some effect – values of “ y ”. The same is not true for $R(x,y)$; here x and y are related but the role of dependent variable is not allocated to any particular variable.

The fundamental operation that makes transformations of fuzzy sets through fuzzy relations possible is a composition (convolution) mechanism. More specifically if R is a fuzzy relation in $X \times Y$ and A is a fuzzy set in X , then a sup- t composition of A and R , denoted by B gives rise to a fuzzy set (B) arising in Y with the following membership function

$$B(y) = (A \circ R)(y) = \sup_{x \in X} [A(x) \wedge R(x, y)] \quad (19)$$

The essence of this composition is illustrated in Figure 16; note that a fuzzy relation R is a granular generalization of a certain function (mapping from X to Y).

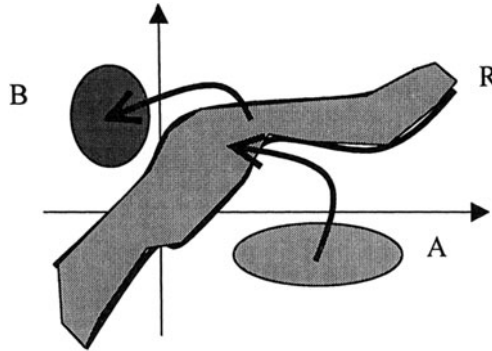


Figure 16. An example of the sup- t composition of A and R .

The above composition subsumes all standard models of transformation of granular or numeric information. A summary of the most representative scenarios is covered in Table 2. The underlying criterion is that of granularity of information either with the entities being processed or/and a vehicle used in their transformation.

Granularity of input and mapping	Function (f)	Relation (R)
Numeric input x	Function transformation resulting in numeric value in Y , $y=f(x)$	Fuzzy set in Y with the membership function equal to $R(x,y)$
Fuzzy set A	Fuzzy set in Y computed	Fuzzy set in Y being a

	with the use of the extension principle	result of the sup-t composition of A and R
--	--	---

Table 2. Examples of transformation between input and output spaces; the distinction between the cases is based upon granularity of available information or mapping mechanism.

It is instructive to note that all the cases in the above table can be derived as special cases of the sup-t composition (shadowed entry).

3. 11 FUZZY SETS AND MULTIVALUED LOGIC

Likewise in set theory where we showed a direct relationship between sets and two-valued logic treating these two constructs as isomorphic structures, the same relationship holds between fuzzy sets and multivalued logic. A truth-value of a proposition in the multivalued logic corresponds to the membership grade of an element to a given fuzzy set. Subsequently the logic *and* and *or* operations in this logic are realized via t- and s-norms in the same way in which we aggregate fuzzy sets. De Morgan laws of fuzzy sets translate directly into the analogous constructs encountered in multivalued logic. Table 3 summarizes these two formalisms in a succinct manner.

Notion	Fuzzy sets	Multivalued logic
Basic entity	fuzzy set	proposition
Basic predicate	Degree of membership	Truth value
Operations	union intersection complement	<i>or</i> operation <i>and</i> operation negation

Table 3. Fuzzy sets and multivalued logic: the basic notions and operations.

The term fuzzy logic being in a common usage (and sometimes treated as a synonym for fuzzy sets) requires some clarification. In essence, fuzzy logic is concerned with fuzzy truth values described by fuzzy sets defined in the unit interval that take on linguistic values such as *true*, *very true*, *more or less true*. In the sequel, the computations of the truth value of any compound statements, say *P and Q* with *P* and *Q* assuming some linguistic truth values we use the standard extension principle. As an example the truth value of the statement *P or Q* (*R*) reads as

$$R(z) = \sup_{u,w \in [0,1]: z=usw} [\min(P(u), Q(w))]$$

$z \in [0,1]$ and the *or* connective is modeled by means of some s-norm and the corresponding truth values of *P* and *Q* are given. In general, we have

$$R(z) = \sup_{u,w \in [0,1]: z=u \phi w} [\min(P(u), Q(w))]$$

where ϕ is a realization of some compound logic statement of its two components P and Q (say *and*, *or*, *exclusive or*, *equivalence*, etc.).

3. 12 CALIBRATION OF FUZZY SETS: ESTIMATION OF MEMBERSHIP FUNCTIONS AND AN ADJUSTMENT OF UNIVERSE OF DISCOURSE

Fuzzy sets are context-driven constructs. While they come with a well-defined semantics and are general in the sense that the same concept can be used across a number of problems. The term *safe* speed does apply to an array of traffic situations (highway, congested street, country road, different weather conditions) and is equally important yet the membership function pertinent to the concept itself needs to be calibrated. This calibration is realized in two different ways. First, the membership function can be adjusted. Second, the universe of discourse can be modified. The calibration of the membership functions is carried out in two ways:

- through the estimation of the individual membership grades for discrete elements of the universe of discourse. This is usually completed through various polling techniques, cf. Pedrycz and Gomide (1989) and Zimmermann (2001). There are a number of specific experimental techniques that help reduce some potential experimental bias:

- through pairwise comparisons. This technique originates from the hierarchical decision processes as discussed by Saaty (1980). To determine the membership grades one carries out a series of pairwise comparisons where we assess preference of two elements of the universe of discourse at a time (the elements are evaluated vis-à-vis a certain concept for which we want to construct a membership function), organize these experimental findings in the form of a so-called reciprocal matrix and develops a membership function as an eigenvector corresponding to the largest eigenvalue. This technique allows us also to identify possible inconsistencies in the data and quantify their level.

The calibration of the universe of discourse is concerned with the development of a certain mapping Γ between the given universe X (over which we have a collection of fuzzy sets) and a new one (X'), say $x'=\Gamma(x)$, Figure 17. As a consequence of this transformation any fuzzy set originally existing in X is transformed as $A'(x')=A(\Gamma(x))$. The mapping could be nonlinear and depending upon the type of this nonlinearity several regions of X are treated differently. It means that some regions are stretched while others undergo a certain contraction. The detailed discussion of the calibration is presented in Pedrycz and Gomide (1997).

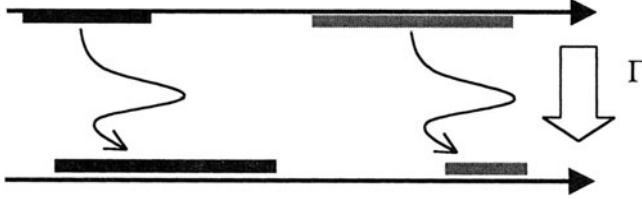


Figure 17. An example of a nonlinear transformation (Γ) of the universe of discourse X .

3. 13 THE EMBEDDING PRINCIPLE

Fuzzy sets form an interesting and useful platform for the realization of a so-called embedding principle. Its crux can be described as follows. Problems involving sets are difficult to optimize because of the discontinuous 0-1 nature of their characteristic functions. Owing to this discontinuity, there is no doubt that gradient-based methods will fail. Essentially, we note that the derivative defined as

$$\frac{\partial A}{\partial p} = \begin{cases} 1 & \text{if } p \in \text{boundary of } A \\ 0 & \text{otherwise} \end{cases}$$

comes with an inherent danger of producing an iterative gradient-based algorithm that easily stacks at any point of the search space. Note that the derivative zeroes in all places but the edge of the set, Figure 18. The likelihood of having no update of the parameters where

$$\mathbf{p} = \mathbf{p} - \alpha \nabla_{\mathbf{p}} A$$

where \mathbf{p} stands for a vector of the parameter to be optimized and α denotes a learning rate is very high. Practically this behavior of the gradient will lead to the collapse of the optimization process.

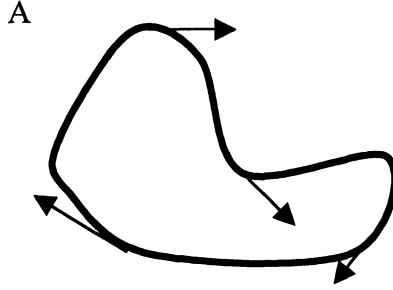


Figure 18. The gradient-based optimization; note that the optimization occurs at the boundary of the set; the arrows visualize the gradient of A .

The idea is to embed the problems involving sets into a framework of fuzzy sets, solve the problem in this generalized setting, move back with the fuzzy set solution to the original search space (this with sets) and convert the fuzzy set into its set-based counterpart, namely $\mu(X) \rightarrow \mathcal{I}(X) \rightarrow \mu(X)$. The three functional phases identified in this embedding process are visualized in Figure 19.

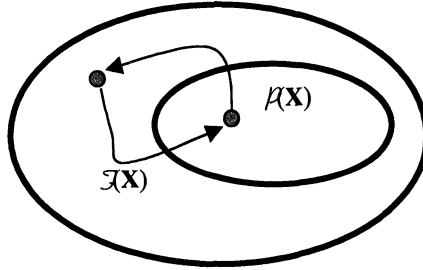


Figure 19. The embedding principle and its three- phase realization: embedding in $\mathcal{I}(X)$, determination of a solution, and its contraction to element in $\mu(X)$.

In a nutshell, we can treat this scheme as an example of communication occurring between granular worlds of fuzzy sets $\mathcal{I}(X)$ and sets $\mu(X)$; this material will be studied in depth in Chapter xx.

3. 14 CONCLUSIONS

As discussed in Klir (2001), fuzzy sets come with four facets, that is set-theoretic, relational, logical, and epistemological. The set-theoretic facet concentrates on the imprecise boundaries, grades of membership. The relational aspects address the

matter of multidimensionality where we look at fuzzy relations and, in essence, consider graded constraints. Fuzzy logic is about the logical facet and revolves around degrees of truth. It is a formalized logic language of building complex propositions on a basis of simple propositions and their truth degrees. The epistemological facet is about evidence, belief and what the degrees of truth are all about.

REFERENCES

- Bargiela, A., Pedrycz, W. (2002), A model of granular data: A design problem with the Tschebyshev-based clustering, *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2002*, Hawai, 578-583.
- Bellman, R., Kalaba, L., Zadeh, L.A. (1966), Abstraction and pattern classification, *J. Math. Anal. Appl.* **13**, 1-7.
- Black, M. (1937), Vagueness: an exercise in logical analysis, *Philosophy of Science*, **4**, 427-455.
- Butnariu, D., Klement, E.P.(1993), Triangular Norm Based Measures and Games with Fuzzy Coalitions, *Kluwer Academic Publishers*, Dordrecht.
- Capocelli, R.M., de Luca, A.(1973), Fuzzy sets and decision theory, *Information & Control*, **23**, 446-473.
- Di Nola, A., Sessa, S., Pedrycz, W., Sanchez, E. (1989), *Fuzzy Relation Equations and Their Applications to Knowledge Engineering*, Kluwer Academic Publishers, Dordercht.
- Dubois, D., Prade, H. (1997), The three semantics of fuzzy sets, *Fuzzy Sets and Systems*, **90**, 141-150.
- Dubois, D., Prade, H. (1980), *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, New York.
- Duhem, P.(1906), *The Aim and Structure of Physical Theory*, Paris.
- Ebanks, E. (1983), On measures of fuzziness and their representations, *J. Math. Anal & Appl.*, **94**, 24-37.
- Gottwald, S. (1979), A note on measures of fuzziness, *EIK*, **15**, 221-223.
- Kandel, A. (1982), *Fuzzy Techniques in Pattern Recognition*, J. Wiley, New York.
- Klement, E.P. Mesiar, R., Pap, E. (2000), *Triangular Norms*, Kluwer Academic Publishers, Dordercht.
- Klir, G.J., Folger, T.A. (1988), *Fuzzy Sets, Uncertainty and Information*, Prentice Hall, Englewood Cliffs.
- Klir, G.J. (2001), Foundations of fuzzy set theory and fuzzy logic: a historical overview, *J. General Systems*, **30(2)**, 91-132.
- Knopfmacher, J. (1975), On measures of fuzziness, *J. Math Anal & Appl.*, **49**, 529-534.

Korzybski, A. (1933), *Science and Sanity*, Int. Non-Aristotelian Library Publishing Company, Lakeville, Conn.

Kosko, B.(1992), *Neural Networks and Fuzzy Systems*, Prentice Hall, Englewood Cliffs.

Miller, G.A. (1956), The magical number seven plus or minus two: some limits on our capacity for processing information, *Psychological Review*, **63**, 81-97.

Pedrycz, W., Bargiela, A. (2001), Information granulation: A search for data structures, *Knowledge-based Engineering Systems KES 2001*, Osaka, October 2001, 1147-1151.

Pedrycz, W., Gomide, F.(1998), *Fuzzy Sets*, MIT Press, Cambridge MA.

Pedrycz, W., Gomide, F. (1997), Nonlinear context adaptation in the calibration of fuzzy sets, *Fuzzy Sets and Systems*, **88**, 91-97.

Saaty, T.L.(1980), *The Analytic Hierarchy Processes*, McGraw Hill, New York.

Schweizer, B., Sklar, A.(1983), *Probabilistic Metric Spaces*, North Holland, Amsterdam.

Trillias, E., Riera, T.(1978), Entropies in finite fuzzy sets, *Information Sciences*, **15**, 159-168.

Wiener, N. (1923), On the nature of mathematical thinking, *Australian J. of Psychology and Philosophy*, **1(4)**, 268-272.

R.R. Yager, R.R. (1983), Entropy and specificity in a mathematical theory of evidence, *Int. J. Gen Systems*, **9**, 249-260.

Zadeh, L.A. (1965), Fuzzy sets, *Information & Control*, **8**, 338-353.

Zadeh,L.A. (1975), The concept of a linguistic variable and its application to approximate reasoning, *Information Sciences*, **8**, 199-249 (part 1) and **9**, 43-80 (part 2).

Zadeh, L.A. (1978), Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems*, **1**, 3-28.

Zimmermann, H.J.(2001), *Fuzzy Set Theory and Its Applications*, 4th edition, Kluwer Academic Publishers, Boston.

ROUGH SETS

4. 1 INTRODUCTION

Rough sets – a concept of granular computing introduced by Zdzislaw Pawlak (Pawlak, 1982, 1984, 1986, 1991, 1999) are concerned about the notion of roughness. It inherently arises when we are interested in describing concepts in the language of some generic knowledge-based entities and is intimately linked with an idea of indiscernibility between elements (more formally, an indiscernibility relation). In other words, we may treat rough sets as a framework in which we represent concepts in the setting of indiscernibility relations. It is convenient to cast rough sets and their underlying methodology in the general two-phase development process as being usually applied in practice. First, we consider (assume) a collection of generic descriptors that are treated as essential blocks whose distinction and usage is central to all cognitive activities going on in the problem under consideration. Second, we express any new concept X (either emerging in the problem or being communicated by the external environment) in the language of the generic descriptors we have identified in the first phase. Evidently, an expression of X carried out by means of these descriptors may not be perfect (and this is usually the case), hence we end up with some “roughness” of the characterization of X .

In this chapter, we concentrate on the underlying idea, present a formalism that supports it and discuss some algorithmic aspects. An abundant literature reports on a diversity of applications and helps develop a global perspective as to the most promising development trends and applications (Lin and Cercone, 1997; Pal and Skowron, 1999; Pawlak et al., 1988; Peters and Ramanna, 1999; Skowron, 1989(a)-(b); Swiniarski, 1999; Ziarko, 1993).

4. 2 THE CONCEPT

Before moving on to the detailed formal environment, we start with two intuitively appealing examples that help us appreciate the formalism of rough sets.

(a) We are interested in the description of a highway traffic. The two essential variables that naturally arise in this context are an average speed of vehicles and their acceleration. Both speed and acceleration are discretized (quantized) in terms of intervals (sets) where each interval is denoted by A_1, A_2, \dots, A_c (speed) and B_1, B_2, \dots, B_p (acceleration). A Cartesian product $A_i \times B_j$ describes some region in the space of the highway traffic. The union of all such Cartesian products builds a vocabulary of terms (descriptors) we will then use in this problem, see Figure 1.

More formally we have $\mathbf{A} = \bigcup_{i,j}^{c,p} (A_i \times B_j)$.

acceleration

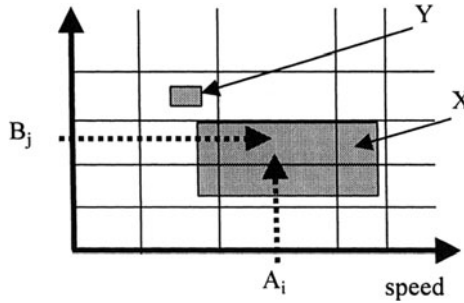


Figure 1. A collection of descriptors of highway traffic in the space of speed and acceleration; a granular observation X to be expressed (characterized) by these descriptors leads to an evident “roughness” of the ensuing characterization. The same does not hold for Y .

An observation of a current traffic situation realized over some period of time produces an information granule $X = [a, b] \times [c, d]$, refer again to Figure 1. Then we represent X in the language of the assumed discretization (namely descriptors of speed and acceleration). Apparently, X does not “fit” ideally any of the elements of \mathbf{A} . Its description can be realized by determining some type of “envelope” and specifying its lower and upper bound (approximation). The upper bound consists of all elements of \mathbf{A} that have something in common with X meaning that their intersection with X is nonempty. Denote the bound by X_+ . The lower bound is tighter and consists of all elements of \mathbf{A} that are fully included in X (so we are confident that X is fully “identified” with them). Here we use the notation X_- . Evidently, through such description in terms of \mathbf{A} , X is located somewhere in-between X_- and X_+ . Several intuitively appealing observations are in place. First, the roughness of the description depends upon some distance between X_+ and X_- . Practically, in case X is of the form shown in Figure 1, it is unlikely to expect that

these two boundaries coincide and result in a complete elimination of the roughness effect. If X becomes far more specific (in comparison with the specificity of \mathbf{A}), this usually reduces the level of roughness. We can note a lower value of roughness for another information granule Y shown in Figure 1.

From a system point of view, one can realize the entire process of describing X in the form visualized in Figure 2: an input X is translated via the elements of \mathbf{A} and gives rise to the lower and upper approximation (X_- , X_+). The two outputs are viewed as the result of such transformation.

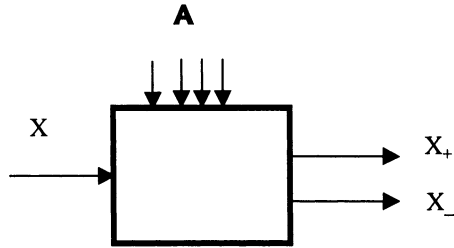


Figure 2. A system view at the characterization of X in the language of the descriptors forming the elements of \mathbf{A} .

(b) The second example is concerned with a classification problem. We are provided with a collection of objects described by a family of features (attributes) that belong to one of the two classes (class ω_1 and ω_0). Each attribute assumes a finite number of values. We are interested in the description of the classes. The simplest way would be to enumerate all patterns (objects) that belong to ω_1 and ω_0 , respectively. This enumeration boils down to the recording of the values of the attributes of the corresponding objects. Alluding to the language of descriptors, we may regard X to be a class descriptor that is built (constructed) on a basis of \mathbf{A} (that in this case is just a collection of the objects). It may also happen that there are some objects for which we have the same values of the attributes but they belong to two different classes. In this case, in the characterization we are faced with an inherent roughness of the class description. In other words, while there are some values of the attributes that point at one specific class, the same values characterize some objects belonging to ω_0 and others that are excluded from it. Figure 3 illustrates the phenomenon of roughness in the characterization of the classes.

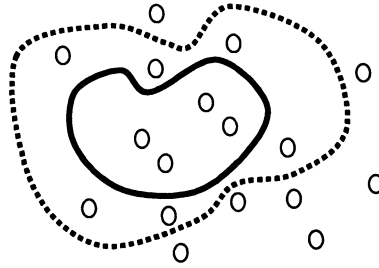


Figure 3. Roughness in class description; shown are objects belonging to the given class (inside solid contour) and excluded from the class (outside dotted contour). The objects within these two lines visualize the roughness of description of the class.

We are now prepared to proceed with a formal definition of rough sets. First we introduce a concept of an information system that plays here a central role.

4. 3 INFORMATION SYSTEMS

An information system S is a formal structure viewed as a four-tuple of the form

$$S = \langle X, Q, V, f \rangle \quad (1)$$

whose components assume the following roles. X is a finite universe including all elements (objects) we are interested in some problem description, $X = \{x_1, x_2, \dots, x_N\}$. Q is a finite set of attributes (features) used in the description of elements of

X ; $Q = \{q_1, q_2, \dots, q_n\}$. V describes values of all attributes, that is $V = \bigcup_{i=1}^n V_i$ with V_i

forming a set of values of the i -th attribute. Finally “ f ” linking all the components we discussed so far is called a decision (information) function and reads as follows

$$f : X \times Q \rightarrow V \quad (2)$$

We have $f(x, i) \in V_i$ where “ i ” denotes an i -th attribute in Q and x is confined to X . Any pair (q, v) , $q \in Q$ and $v \in V$ is called a description of the information system.

Owing to the finite number of objects and attributes, it is convenient to treat an information system as a certain data table in which rows (N) relate to the corresponding objects and columns include the values of the successive attributes (n). Thus each row describes a certain object.

Alluding to the classification example in the previous section, we can collect and represent information about the objects in the data table given in Table 1 (here we consider only three attributes assuming values from $\{0,1,2\}$, $\{L, H\}$, and $\{0, 1\}$, respectively). Using the previous notation, we have $X = \{x_1, x_2, \dots, x_{10}\}$, $V_1 = \{0, 1, 2\}$, $V_2 = \{L, H\}$ and $V_3 = \{0, 1\}$. What is usual in any classification problem, one (or more) attribute plays a role of a label that is an indicator pointing to which class (group) a certain object belongs. In this example, the last column (attribute) in the table assumes this role and it assigns (discriminates) the objects into the two classes (class 0 and class 1).

object	Attribute 1	Attribute 2	Attribute 3
x_1	0	L	0
x_2	0	H	0
x_3	0	H	0
x_4	1	L	0
x_5	1	L	1
x_6	1	H	1
x_7	1	H	1
x_8	2	L	0
x_9	2	L	1
x_{10}	2	H	1

Table 1. An example of data (information) table with three attributes.

The fundamental notion that bears an evident practical relevance is the one of indiscernibility. With the same information system S in mind, let us denote by A a subset of attributes, $A \subseteq Q$. We say that two objects (x and y) are indiscernible by the set of attributes A in S if and only if (iff) $f(x,a) = f(y,a)$ for every a in A . This observation about indiscernibility is denoted by xAy . In essence, what this relationship does, it forms an equivalence relation (indiscernibility) in X by bringing together all objects that are not distinguishable in the selected subset of the attributes A ,

$$\text{IND}(A) = \{ (x, y) \in X \times X \mid f(x, a) = f(y, a) \text{ for all } a \in A \} \quad (3)$$

As the objects which satisfy the above relation cannot be distinguished one from another (so they are equivalent), we can form a notion of an equivalence class $[x]_A$ induced with respect to A ,

$$[x]_A = \{ y \in X \mid (x, y) \in \text{IND}(A) \} \quad (4)$$

We also say that $[x]_A$ is an A-elementary set (equivalence class) including x. An ordered pair $AS = (X, IND(A))$ is called an approximation space. Again, we stress that such approximation space is implied by the selected subset of the attributes A. Different As may lead to different approximation spaces.

Returning to the data shown in Table 1, and depending upon the choice of A we generate different equivalence classes.

For $A = \{1, 2\}$ that takes into consideration two first attributes, we have

$$\begin{aligned} [x_1]_A &= \{x_1\} \\ [x_2]_A &= [x_3]_A = \{x_2, x_3\} \\ [x_4]_A &= [x_5]_A = \{x_4, x_5\} \\ [x_6]_A &= [x_7]_A = \{x_6, x_7\} \\ [x_8]_A &= [x_9]_A = \{x_8, x_9\} \\ [x_{10}]_A &= \{x_{10}\} \end{aligned}$$

For $A = \{1\}$ we get

$$\begin{aligned} [x_1]_A &= [x_2]_A = [x_3]_A = \{x_1, x_2, x_3\} \\ [x_4]_A &= [x_5]_A = [x_6]_A = [x_7]_A = \{x_4, x_5, x_6, x_7\} \\ [x_8]_A &= [x_9]_A = [x_{10}]_A = \{x_8, x_9, x_{10}\} \end{aligned}$$

For $A = \{2\}$ one has the equivalence classes

$$\begin{aligned} [x_1]_A &= [x_4]_A = [x_5]_A = [x_8]_A = [x_9]_A = \{x_1, x_4, x_5, x_8, x_9\} \\ [x_2]_A &= [x_3]_A = [x_6]_A = [x_7]_A = [x_{10}]_A = \{x_2, x_3, x_6, x_7, x_{10}\} \end{aligned}$$

Finally, for $A = \{3\}$ (in which case we are interested in the equivalence relation being induced by the assigned classes – note that this attribute deals with two classes, 0 and 1) we obtain

$$\begin{aligned} [x_1]_A &= [x_2]_A = [x_3]_A = [x_4]_A = [x_8]_A = \{x_1, x_2, x_3, x_4, x_8\} \\ [x_5]_A &= [x_6]_A = [x_7]_A = [x_9]_A = [x_{10}]_A = \{x_5, x_6, x_7, x_9, x_{10}\} \end{aligned}$$

If the attribute set V is divided into two disjoint sets such called condition (C) and decision attributes (D), then such information systems are referred to as decision tables,

$$S = \langle X, Q, C \cup D, f \rangle \quad (5)$$

Obviously we have

$$C \cup D = V, \quad C \cap D = \emptyset$$

Conceptually, the previous information systems do not exhibit any “directionality” that is all attributes play the same role. In decision tables we encounter an evident “directionality” component that is we can distinguish between attributes that impact others (namely condition attributes imply some values of the decision attributes). In this sense we can treat decision tables as special cases of the information systems with some directionality constraint. To draw some pertinent analogy, we may contrast relations and functions: relations are direction-free (as we are talking about a notion of “being related”) while functions are directional constructs (as we clearly distinguish between independent and dependent variables). From the basis of this background structure we can introduce a formal construct of a rough set.

4. 4 ROUGH SETS AS SET APPROXIMATIONS

As before we consider an information system S and a subset of attributes A implying a certain approximation space $AS = (X, IND(A))$. We are interested in describing a concept that is a subset X of object in X , $X \subset X$ by making use of the attributes in A (that is considering the given approximation space AS). We introduce two fundamental notions

- lower approximation of X in AS (or A -lower approximation, to emphasize the role played here by A)

$$X_- = \{x \in X \mid [x]_A \subseteq X\} \quad (6)$$

- upper approximation of X in AS (A -upper approximation)

$$X_+ = \{x \in X \mid [x]_A \cap X \neq \emptyset\} \quad (7)$$

We say that AX_- and AX_+ (or briefly X_- and X_+) are approximations of concept X in AS . The essence of these approximations becomes evident by analyzing the way in which X is expressed by means of the equivalence classes. The lower approximation is a conservative construct as we admit all $[x]_A$ s that are fully included (contained) in X . The upper approximation is far more liberal as we admit equivalence classes once they intersect with X .

In a nutshell, a rough set is a construct represented by means of two approximations, $\langle X_-, X_+ \rangle$. We would like to stress that rough sets are defined vis-à-vis a certain approximation space. The selection of the approximation space implies the definition of the rough set. In other words, the same concept X may lead to different descriptions depending upon the assumed approximation space AS .

As the first example, let us refer to the traffic description (Figure 1) and consider a concept X shown by the shadowed area. The approximation space is built by the Cartesian products of intervals (sets) A_i and B_j (in the notation we have used so far, these products are equivalence classes standing in the fundamental definition of rough sets). The lower approximation consists of a single Cartesian product

$$X_- = \{A_i \times B_j\}$$

The upper approximation is built out of the Cartesian products that overlap (intersect) with X ; those are $A_i \times B_j$, $A_{i-1} \times B_j$, $A_{i-1} \times B_{j-1}$, $A_{i+1} \times B_{j-1}$, $A_{i+1} \times B_j$, $A_i \times B_{j-1}$ hence we get

$$X_+ = \{A_i \times B_j, A_{i-1} \times B_j, A_{i-1} \times B_{j-1}, A_{i+1} \times B_{j-1}, A_{i+1} \times B_j, A_i \times B_{j-1}\}$$

Immediately we can note that $X_- \subset X_+$ with these two approximations being quite different. Considering again the data in Table 1 and treating the third attribute as a decision attribute, the concept X (class 1) includes a subset of X ,

$$X = \{x_5, x_6, x_7, x_9, x_{10}\}$$

With the approximation space built by the two attributes, we have the following approximations

$$\begin{aligned} X_- &= \{x_6, x_7, x_{10}\} \\ X_+ &= \{x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\} \end{aligned}$$

We note that different approximation spaces produce different rough sets that are manifestations of the same concept. What becomes of particular interest is a difference between the lower and upper approximation that profoundly tells us about the roughness of the description resulting from the given AS.

In the sequel we discuss properties and some quantitative characterization of aspects of rough sets, especially those dealing with the way how we quantify a notion of roughness of these information granules.

4.5 CHARACTERIZATION OF ROUGH SETS

Rough sets capture the description of any given concept in the setting of a certain approximation space. It became obvious from the observations made in the previous sections that rough sets could be different with respect to their roughness. Intuitively, it makes sense to quantify how rough a given rough set is. The essence of rough sets is conveyed by the lower and upper approximations. The difference

between them comes under a name of a boundary region (A-boundary region, to be more detailed) that is computed in the form

$$B(X) = X_- \setminus X_+ \quad (8)$$

and deals with all elements of S which cannot be classified as belonging to the concept or excluded from it.

In the two examples studied above, we have the corresponding boundaries: for the highway example

$$B(X) = \{A_{i-1} \times B_{j-1}, A_{i-1} \times B_j, A_i \times B_{j-1}, A_{i+1} \times B_{j-1}, A_{i+1} \times B_j\}$$

while for the classification problem the boundary is equal to $\{x_4, x_5, x_8, x_9\}$. The accuracy of approximation of X by the set of attributes A (accuracy, for short) is a scalar measure describing how rough the given rough set is defined as the following ratio involving the cardinalities of the approximations

$$\alpha(X) = \frac{\text{card}(X_-)}{\text{card}(X_+)} \quad (9)$$

If X is A -exactly approximated in AS then $\alpha(X) = 1$. The lower the value of $\alpha(X)$, the higher the roughness of X .

Returning to the two examples studied above, we have the roughness of the rough sets equal to $1/6$ and $3/7$, respectively.

The main properties of rough sets that are articulated in terms of their approximations are summarized in Table 2.

$$\begin{aligned} X_- &\subseteq X \subseteq X_+ \\ \emptyset_- &= \emptyset_+ = \emptyset \\ X_- &= X_+ = X \\ (X \cup Y)_- &\supseteq X_- \cup Y_- \\ (X \cup Y)_+ &= X_+ \cup Y_+ \\ (X \cap Y)_- &= X_- \cap Y_- \\ (X \cap Y)_+ &\subseteq X_+ \cap Y_+ \\ (X)_- &= \setminus (X_+) \\ (X)_+ &= \setminus (X_-) \end{aligned}$$

Table 2. Main properties of rough sets (\setminus denotes a set complement).

4. 6 SET COMPARISONS IN THE SETTING OF ROUGH SETS

It is obvious how to compare sets and the definition of set equality is intuitive: we consider two sets to be equal if they consist of the same elements. In rough sets, a definition of equality needs to be revisited as we may have two sets that are not equal (in a usual sense) yet they can be deemed equal when being discussed in the context of rough sets (because of the existence of the approximation bounds). Interestingly, we end up with three different definition of equality (Pawlak, 1991)

Bottom A – equality:

Sets X and Y are bottom A – equal (denoted by $X =_- Y$) if $X_- = Y_-$.

Top A-equality:

Sets X and Y are top A – equal (denoted by $X =_+ Y$) if $X_+ = Y_+$.

A-equality:

Sets X and Y are A – equal (denoted by $X = Y$) if $X_- = Y_-$ and $X_+ = Y_+$.

The latter definition is the strongest in the sense we require that the equality holds with respect to both approximations. Several examples illustrating the above definitions are shown in Figure 4. When dealing with classification problems, these definitions come with different interpretations. Bottom A-equality states that the positive examples of X and Y are the same (the sets of negative examples could be different). The opposite situation holds for top A-equality where we see negative examples represented by X and Y to be equal. Both positive and negative examples are the same when we confine ourselves to the definition of the A-equality.

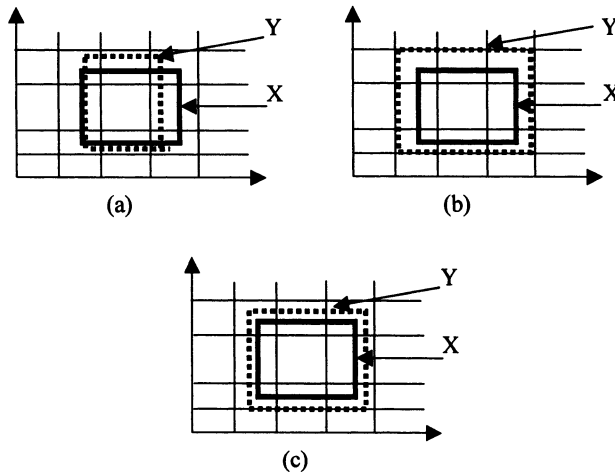


Figure 4. Examples of rough equality of sets: bottom A-equality (a), top A-equality (b), A-equality (c).

In the same way we define rough inclusion of sets, namely

Bottom A – inclusion:

Set X bottom A – included in Y (denoted by $X \subset_- Y$) if $X_- \subset Y_-$.

Top A-inclusion:

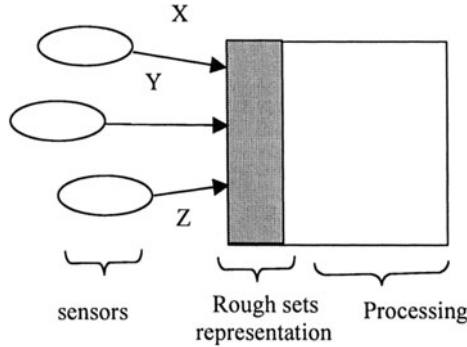
Set X is top A – included in Y (denoted by $X \subset_+ Y$) if $X_+ \subset Y_+$.

A-inclusion:

Set X is A – included in Y (denoted by $X \subset Y$) if $X_- \subset Y_-$ and $X_+ \subset Y_+$.

It should be stressed that neither rough equality nor rough inclusion implies equality or inclusion in a usual set-theoretic sense.

In what follows, we discuss how operations on rough sets and their comparison applies to problems of sensor fusion and communication problems (this matter will be discussed in great detail in Part III of the book). A general architecture of the granular environment in which processing takes place is visualized in Figure 5.



(a)

Figure 5(a). A general architecture of sensor fusion.

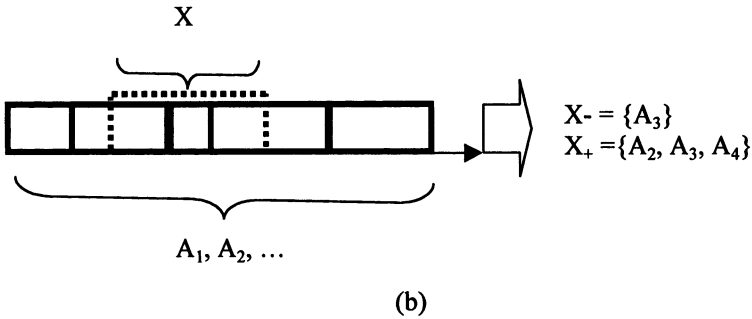


Figure 5(b). Representation of sensor reading X resulting in a rough set (b).

The central part of the overall model is an interface that links sensors' readings with its processing part. The interface is formed by a collection of intervals which are just equivalence classes; denote them by A_1, A_2, \dots, A_c . As seen, each of them is an interval in the line of reals, \mathbf{R} . Each sensor produces an interval type of information, say X, Y, Z , etc where each of them is an interval in \mathbf{R} . This type of granular information could be accessible when we observe some variable over some period of time and form an interval of the values captured by the sensor during this time window. Inevitably, the values of the variable can fluctuate and this gives rise to the interval-type of information. On top of this, we may also envision a limited accuracy of the sensor that translates into intervals of possible values. Expressing the input from the sensors leads to some rough sets; it is apparent that X translates to the lower and upper approximation, see Figure 5(b). The lower and upper approximations of the readings of the sensor (X, Y, Z, U , etc) are "visible" to the following functional module of the scheme. Similarly, we can assess the readings of the sensor in terms of their inclusion and equality ($X \subset Y, X = Y$, etc.). Similarly, we can complete some aggregation of the sensors by completing their union and intersection and express the result in the language of the equivalence classes (A_1, A_2 , etc.). It is noticeable that the processing occurring at the later part of the architecture operates on the equivalence classes in spite of the variability of the existing sensor information.

4. 7 REDUCTION OF ATTRIBUTE SPACES AND REDUCTS

As we have already noticed different approximation spaces lead to different rough sets whose degrees of roughness could vary quite substantially. In practice we are commonly interested in a small set of attributes hoping that some of them may be deemed irrelevant or marginal to the quality of the classification results (in other

words, we anticipate that there is a core collection of the attributes that make the job). The process of finding a small number of attributes (a subset of the original family of the attributes) is referred to as attribute reduction. This reduction process leads us to the notion of a reduct. By which we mean an essential subset of attributes that can discern all objects that are discernable by the original complete set of attributes. During the process of constructing a reduct we eliminate redundant (superfluous) attributes (obviously this reduction pertains to a specific classification process).

Considering information system S with a subset of attributes $A \subset Q$, we say that attribute $a \in A$ is dispensable in A if the respective indiscernibility relations are equal, namely $IND(A) = IND(A \setminus \{a\})$. If this relationship does not hold, we call a to be indispensable. The set of all indispensable attributes of A is called a core of A (denoted by $core(A)$) is concerned with such attributes that cannot be removed from A without causing any loss in the quality of classification. We say that E is a reduct of A , $E = RED(A)$ if E is a minimal set of attributes that discerns all objects in S that are discernable by A and cannot be further reduced. The intersection of all reducts of A is a core of A

$$CORE(A) = \bigcap_{\text{all reducts}} RED(A) \quad (10)$$

It is worth stressing that computing of reducts is an NP-hard problem (Pal and Skowron, 1999) and when dealing with them we usually have to invoke some heuristic methods including evolutionary computing.

4. 8 ROUGH FUNCTIONS

So far, we have concentrated on rough sets as useful models of concepts (descriptors). Rough functions generalize functions and in this way support a transformation of some input data (Pawlak, 1999; Lin, 1997). Functions are self-evident notions. To generalize them to rough functions, we start with a collection of equivalence classes in the input and output variable (domain and co-domain) as visualized in Figure 6.

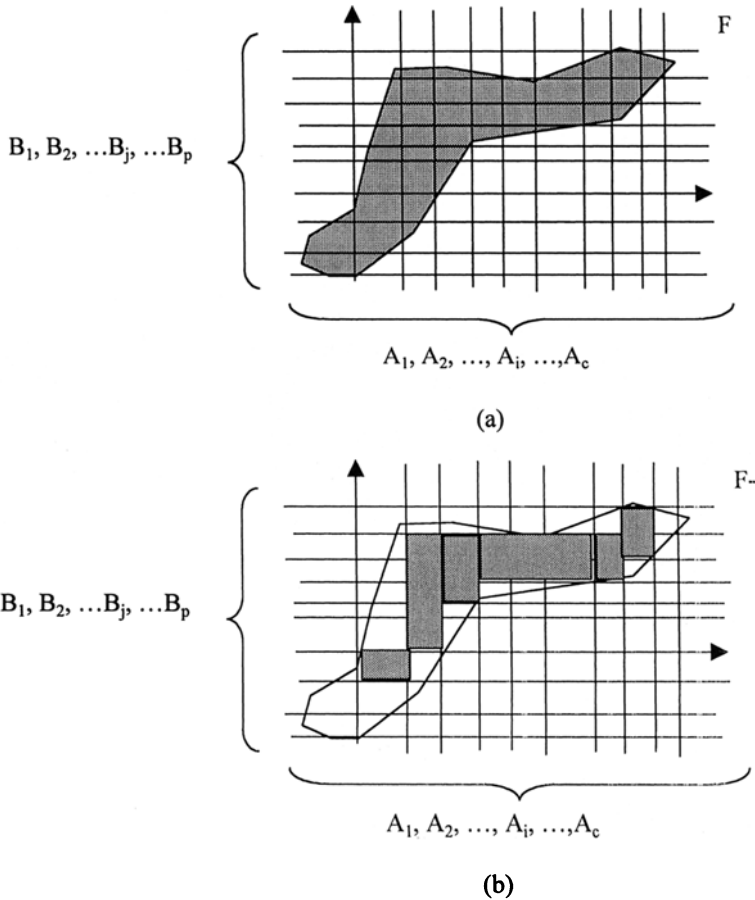


Figure 6. Example of a granular function - interval-valued realization (a) and its rough set model, shown is only its lower approximation F_- . (b).

A granular function can be portrayed as a belt distributed throughout Cartesian products of A_i and B_j . As a matter of fact, it could be viewed as a certain relation. From a technical standpoint, we may think of such granular function as a result of several realizations of some specific numeric function. For instance, in ECG signal analysis we deal with normal QRS segments of the cardiologic signal. These are very characteristic parts of the signal that carry a lot of diagnostic information. However not all normal QRS segments are the same and some variability is present. If we superimpose these signals one on another, we come up with a “thick line” that is a relation rather than a function. This effect is shown in Figure 6(a).

With each A_i we associate a rough set of the values of the function and in this way we can write the function down in the form

$$\langle A_i, (F(A_i)_-, F(A_i)_+) \rangle \quad i=1,2, \dots, c \quad (11)$$

where $(F(A_i)_-, F(A_i)_+)$ is a rough set associated with A_i (that is a rough set indexed by A_i s). The representation shown in the above form will be referred to as a rough function F . We can also use an abbreviated notation by concentrating on the lower and upper approximation and drop the index (i) that finally leads us to the notation (F_-, F_+) . Obviously, one should have (11) in mind to fully appreciate the shorthand notation being used in the above sense.

An interesting operational issue arises on how to transform (map) data and information granules through the rough function. We start with the simplest possible option that is a single numeric datum (real number) $\{x\} \in \mathbf{R}$. The transformation of $\{x\}$ through (F_-, F_+) proceeds in two steps: (a) first we identify an equivalence class A_i which the input $\{x\}$ falls into, and (b) second, we point at the rough set associated with it, that is $(F(A_i)_-, F(A_i)_+)$. For the interval-valued input X , the transformation comprises of two steps. First X is translated into the corresponding rough set (X_-, X_+) . Next we build a lower and upper approximation of the mapping by taking intersections of X_- and F_- as well as X_+ and F_+ , refer to Figure 7.

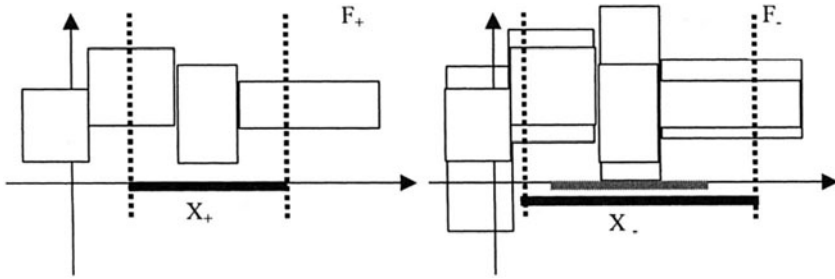


Figure 7. Examples of constructing lower and upper approximations of the transformation of X through a rough function.

4.9 CONCLUSIONS

Rough sets are information granules capable of capturing the effect of roughness arising when dealing with the concepts in the setting of some generic information granules. These granules are a result of the introduction of some indiscernibility relation (equivalence classes). The construct is based upon concept representation. It

becomes evident that an exact representation is not a feasible option (both practically and conceptually) therefore one needs to look at some viable approximation. This is really what is realized by means of rough sets using which we construct the concept approximation by constructing its the lower and upper approximations (bounds). At the same time the roughness of the approximation (which is the crux of any rough set) comes into play and becomes expressed by stating how much the lower and upper approximations differ from each other. The formalism of rough sets dwells on the notion of indiscernibility and all constructs presented in this chapter use this concept in one way or another. Within a formal model of information systems, we are capable of carrying out analysis of attributes and propose fundamental notions dealing with their subsets such as reducts and cores. We showed a number of illustrative examples that help envision a nature of the problems in which rough sets arise in an intuitive fashion. In this applied realm, classification and concept formation tasks are highly dominant. The communication mechanisms are also well supported by rough sets.

REFERENCES

- Lin, T.Y., Cercone, N. (eds.) (1997), *Rough Sets and Data Mining*, Kluwer Academic Publishers, Boston.
- Lin, T.Y.,(1997), Fuzzy controllers: an integrated approach based on fuzzy logic, rough sets, and evolutionary computing, In: Lin, T.Y., Cercone, N. (eds.), *Rough Sets and Data Mining*, Kluwer Academic Publishers, Boston, 123-138.
- Pal, S.K., Skowron A.(eds) (1999), *Rough Fuzzy Hybridization. A New Trend in Decision-Making*, Springer-Verlag, Singapore.
- Pawlak, Z. (1982), Rough sets, *Int. J. of Computer and Information Sciences*, **11**, 341-356.
- Pawlak, Z. (1984), Rough probability, *Bull Pol. Acad. Sci., Math.*, **132**, 607-612.
- Pawlak, Z. (1986), Rough sets and decision tables, *Bull Pol. Acad. Sci., Tech.*, **34**, 563-572.
- Pawlak, Z. (1991), *Rough Sets. Theoretical Aspects of Reasoning About Data*, Kluwer Academic Publishers, Dordrecht.
- Pawlak, Z. (1999), Rough sets, rough function and rough calculus, In: Pal, S.K., Skowron A.(eds), *Rough Fuzzy Hybridization. A New Trend in Decision-Making*, Springer-Verlag, Singapore, 99-109.
- Pawlak, Z., Wong, S.K.M, Ziarko, W. (1988), Rough sets: probabilistic versus deterministic approach, *Int. J. Man-Machine Studies*, **29**, 81-95.
- Peters, J.F., Ramanna, S. (1999), A rough sets approach to assessing software quality: concepts and rough Petri net models, In: Pal, S.K., Skowron A.(eds), *Rough Fuzzy Hybridization. A New Trend in Decision-Making*, Springer-Verlag, Singapore, 349-380.

Skowron, A. (1989a), The relationship between the rough set theory and evidence theory, *Bull Pol. Acad. Sci., Tech.*, **37**, 87-90.

Skowron A. (1989b), Rough decision problems in information systems, *Bull Pol. Acad. Sci., Tech.*, **37**, 59-66.

Swiniarski, R. (1999), Rough sets and principal component analysis and their applications in data model building and classification, In: Pal, S.K., Skowron A.(eds), *Rough Fuzzy Hybridization. A New Trend in Decision-Making*, Springer-Verlag, Singapore, 275-300.

Ziarko, W. (1993), Variable precision rough set model, *J. of Computer and System Sciences*, **46**, 39-59.

GENERALIZATIONS OF INFORMATION GRANULES

In this chapter, we discuss various extensions of the fundamental formal environments of granular computing and elaborate on a series of synergistic interactions arising between them. The first trend is motivated by the complexity and diversity of the granular aspect of information. The second category of developments is motivated by a multifaceted nature of information granularity that usually embraces several dimensions of such concept.

5. 1 INTERVAL-VALUED FUZZY SETS

When capturing a notion of information granularity in terms of fuzzy sets, an interesting question arises with regard to the assignment of single membership values that can be treated as a sound model of membership of the given fuzzy set. It may be argued that as we are dealing with a notion of partial membership, it does not seem justifiable to use a single numeric membership value. Instead, it sounds more reasonable to consider a granular model of the notion of membership. In the simplest form, we can envision a range of possible membership values, that is view some numeric intervals located in $[0,1]$ as legitimate descriptors of feasible membership values. This leads us to the generalized version of fuzzy sets usually referred to as an interval-valued fuzzy sets (Zimmermann, 2001). More formally, an interval- valued fuzzy set (called also Φ -fuzzy set) is a mapping

$$A: X \rightarrow \mathcal{I}([0,1]) \quad (1)$$

For each $x \in X$ we have an interval of membership grades contained in $[0,1]$. Let us recall that $\mathcal{I}(\cdot)$ denotes a family of sets (intervals) defined over $[0,1]$. Effectively, we can describe an interval fuzzy set by its lower and upper bound so that A is fully defined in the form of the following pair

$$A = \langle A_1, A_2 \rangle \quad (2)$$

with A_1 and A_2 denoting a lower and upper bound, respectively. The length of A (viz. a difference between the upper and lower bound) reflects a level of uncertainty (hesitation) associated with each element of the universe of discourse. An example of an interval valued fuzzy set is shown in Figure 1.

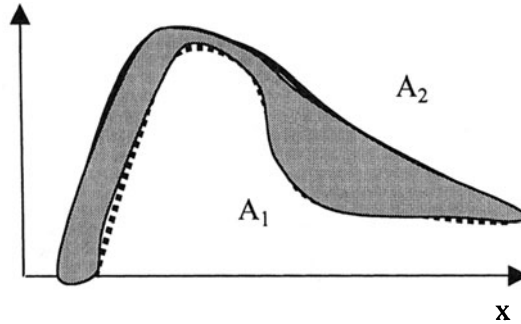


Figure 1. An example of an interval-valued fuzzy set.

In this case the uncertainty (granularity) reflected in the membership grades is higher for lower membership values. For any fuzzy set the boundaries coincide, that is $A_1 = A_2$. In other words, fuzzy sets can be treated as special cases of interval-valued fuzzy sets. The interval nature of the membership grades is found useful in reliable processing of fuzzy sets, cf. Chen et al. (1997), Gorzalczyński (1987), in particular when dealing with propagation schemes of granular data such as those encountered in approximate reasoning.

Computations with interval-valued fuzzy sets follow the fundamental mechanisms encountered in fuzzy sets. The only difference now is that these calculations are carried out for the lower (A_1) and upper (A_2) bound separately. The basic principle of propagation of uncertainty existing in interval analysis becomes here the underlying processing mechanism. Consider two general cases:

- given is a fuzzy relation R and interval-valued fuzzy set $A = \langle A_1, A_2 \rangle$. For these we compute two compositions involving the bounds of A that is $A_1 \circ R$ and $A_2 \circ R$. Hence the uncertainty in the result $\langle A_1 \circ R, A_2 \circ R \rangle$ is a direct consequence implied by the uncertainty (width of the interval of membership grades) residing within A .
- given are an interval-valued fuzzy relation $R = \langle R_1, R_2 \rangle$ and an interval-valued fuzzy set $A = \langle A_1, A_2 \rangle$. In this scenario, the result is an interval-valued fuzzy set whose bounds are determined in the form $B = \langle B_1, B_2 \rangle$ with $B_1 = A_1 \circ R_1$ and $B_2 = A_2 \circ R_2$. Obviously, in this case the boundaries of the resulting interval-valued fuzzy set are broader than in the first case. This is a combined effect of processing granularity of membership values associated with the relation and the interval-valued fuzzy set.

5.2 FUZZY SETS OF TYPE-2 AND HIGHER ORDERS

Fuzzy sets of type 2 (Zimmermann, 2001; Karnik and Mendel, 2001; Mendel and John, 2002) generalize the generic concept of fuzzy sets and are defined as follows

$$A: X \rightarrow \mathcal{F}[0,1] \quad (3)$$

This transformation means that for each element “ x ” in the universe of discourse X we have a fuzzy set of membership grades defined in the unit interval (recall again that $\mathcal{F}(U)$ stands for a family of fuzzy sets defined over U). Having this in mind, we can treat A as a function of two variables: the first one that is associated with the universe of discourse (denoted here by x) and the second one that is defined over the space of membership grades, $u \in [0,1]$. Thus we treat A as a function of two arguments, $A(x,u)$, $x \in X$, $u \in [0,1]$. The shadowed region in Figure 2 is referred to as a footprint of uncertainty and it helps reflect an amount of uncertainty conveyed by this information granule.

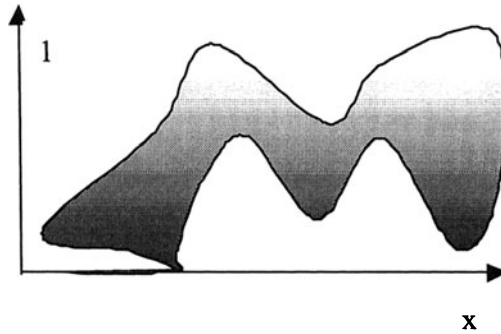


Figure 2. An example of a fuzzy set of type 2.

As the grades of membership are fuzzy sets themselves, one can view fuzzy sets of type 2 as fuzzy sets with linguistic membership grades. An example deals with the type-2 fuzzy set $A = [\text{low low high medium}]$ where low, medium, and high are all fuzzy sets defined in the unit interval with the membership functions, $\text{low}(u) = 1-u$, $\text{medium}(u) = 4u(1-u)$, $\text{high}(u) = u$.

As an example, let us consider the following construction of the fuzzy set of type 2. We define its “skeleton” by providing a membership function

$$A_1(x) = \exp(-(x-m)^2/\sigma^2)$$

and then associate linguistic membership grades with each element of X that are described in the form of the Gaussian membership functions with the modal value equal to $A_1(x)$ and spread μ ,

$$A_2(u) = \text{trunc}[\exp(-(u-A_1(x))^2/\mu^2)]$$

(the role of the truncation function, $\text{trunc}(\cdot)$, is to clip the values of the exponential function to the unit interval). The membership function of the type 2 fuzzy set is described as $A(x,u) = A_1(x)A_2(u)$. Two examples of this fuzzy set are shown in Figure 3. The difference is in the level of uncertainty (size of the footprint of uncertainty) coming with the membership grades. The modal value of the membership function in both cases is equal to 2.

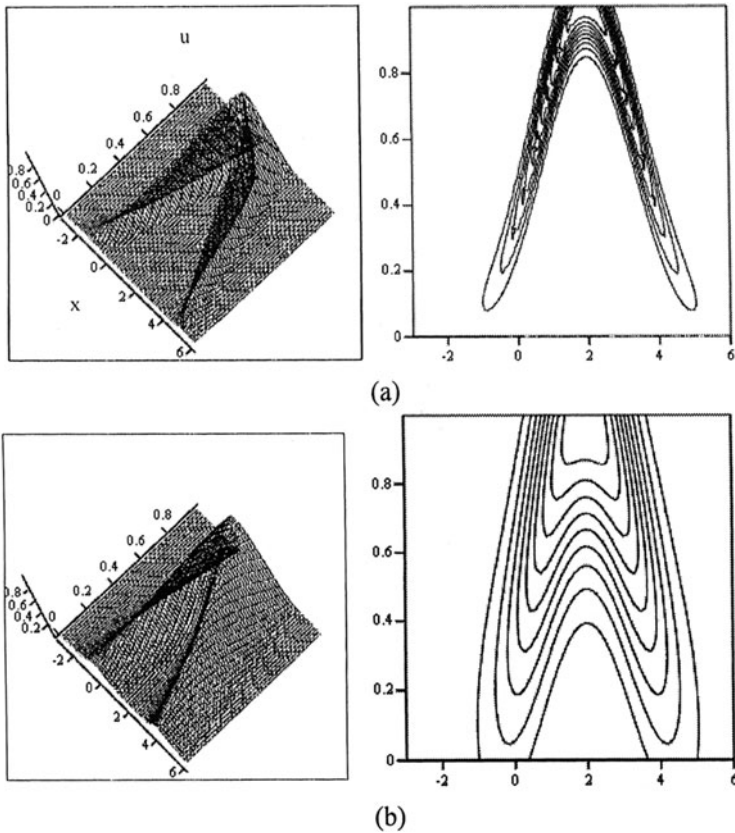


Figure 3. 3D plot and 2D contour display of the fuzzy set of type 2: $m=2$, $\sigma=2$, $\mu=0.1$ (a) and $m=2$, $\sigma=2$, $\mu=0.4$ (b).

This notion can be easily generalized by admitting fuzzy sets of type “k” where the definition is formed by induction that is

$$A: \mathbf{X} \rightarrow \mathcal{I}(\mathcal{I}^{k-1}([0,1])) \quad (4)$$

where $\mathcal{I}(\mathcal{I}^{k-1}(.))$ denotes a family of all fuzzy sets of type “k-1”.

The operations on type 2 fuzzy sets are defined through the extension principle (Zimmermann, 2001). It is easy to note that for each fixed value of x , we end up with two fuzzy sets A_x and B_x (the index is used here to emphasize this specific value of the first argument). Then we have $C_x = A_x \cap B_x$ and $D_x = A_x \cup B_x$ with the following membership functions

$$C_x(w) = \sup_{u,z \in [0,1]: w=utz} [\min(A_x(u), B_x(z))]$$

$$D_x(w) = \sup_{u,z \in [0,1]: w=usz} [\min(A_x(u), B_x(z))]$$

$w \in [0,1]$ where the models of intersection and union are realized through some t - and s -norms, respectively.

The detailed discussion on the operations on type-2 fuzzy sets and their efficient computing is covered in Mendel and John (2002).

5.3 FUZZY SETS OF LEVEL 2 AND HIGHER

Fuzzy sets of level 2 are defined in the form

$$A: \mathcal{I}(\mathbf{X}) \rightarrow [0,1] \quad (5)$$

that is a fuzzy set of level 2 formed over a family of fuzzy sets $\mathcal{I}(\mathbf{X})$, each of which is expressed in \mathbf{X} . The underlying motivation is to develop a structure that is able to represent a higher-level concept by building it by means of a series of low-level concepts. All of them are treated as some entities of the natural language.

As an example, consider a concept of *comfortable* temperature when the acceptability is expressed in the language of linguistic terms of temperature such as *positive small*, *positive medium*, etc. This gives rise to a fuzzy set of type 2, see Figure 4, which itself manifests as a hierarchy of the concepts in which a high-level concept (comfortable temperature) is expressed by means of the low level constructs

(namely the linguistic values of the temperature). More concisely, we can write down A as a vector of membership grades, $A = [0.2 \ 0.5 \ 1.0 \ 0.7 \ 0.3]$ remembering that each of these numeric values refers to the membership functions of the low level concept.

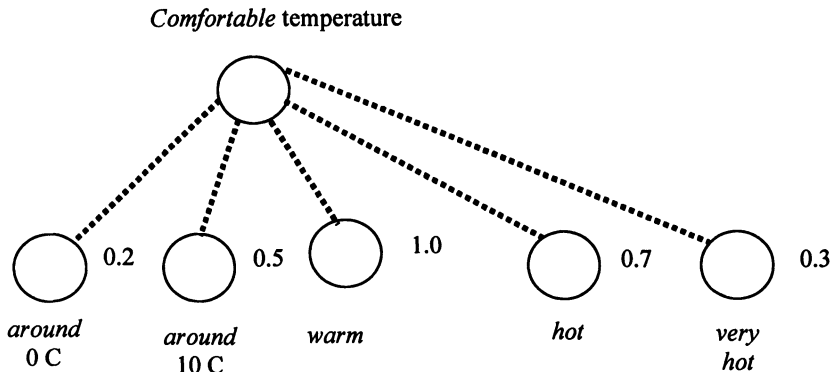


Figure 4. An example of a fuzzy set of level 2.

As in case of type- n fuzzy sets, we can generalize this concept to fuzzy sets of higher level.

5. 4 FUZZY SETS AND ROUGH SETS

Fuzzy sets and rough sets are often considered together and this synergy arises quite naturally (Dubois and Prade, 1992). Three main developments are envisioned in relation to the granular constructs being involved in the synergy.

Let us recall that rough sets are expressed in terms of the indiscernability relation (Pawlak, 1982). In the original definition of rough sets, the elements of the space over which the relation has been defined are sets (or relations). The calculations of the lower and upper bounds for an information granule X are completed using the following expressions for the lower and upper bound

$$X_+ = \{A_i \mid X \subset A_i\} \quad (6)$$

$$X_- = \{A_i \mid X \cap A_i \neq \emptyset\} \quad (7)$$

Fuzzy sets (relations) represented in the setting of sets Consider X is a fuzzy set while A_i are the granular elements (relations) represented as two-valued binary models. The plot of the construct is shown in Figure 5; note that the fuzzy set is visualized as a shadowed spherical object being “dropped” on the regular grid formed by A_i s.

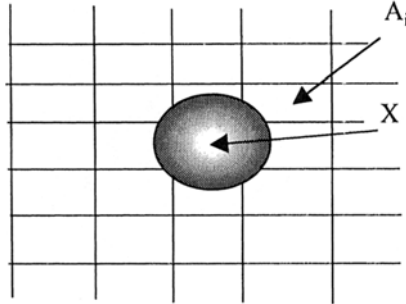


Figure 5. Fuzzy set X expressed in terms of Boolean relations $\{A_i\}$.

An example illustrating this scenario is as follows. The two variables in Figure 5 concerns speed of a car and a fuel consumption. Both these variables (attributes) are granulated by means of intervals, say speed = $\{[10, 20], [20, 40], [40, 60], [60, 80], [80, 120]\}$ (in km/hr), fuel consumption = $\{[1, 5], [5, 7], [7, 12], [12, 16]\}$ (in liters/100 km). Each A_i is a Cartesian product of the intervals coming from the two collections of the information granules, say $A_i = [10, 20] \times [80, 120]$. The concept X to be described is a fuzzy relation $X = \text{medium speed} \times \text{low fuel consumption}$. In pursuing detailed computations, we realize that X_+ and X_- are fuzzy relations rather than collections of the Cartesian products. We reformulate the two previous expressions for the bounds (6) - (7) as follows

$$X_+(A_i) = \inf_{x,y} [\max((1 - A_i(x, y)), X(x, y))]$$

and

$$X_-(A_i) = \sup_{x,y} [\min(X(x, y), A_i(x, y))]$$

so they specify the degrees to which A_i s contribute to the bounds of the descriptors of the concept; the interpretation is the same as in case of sets and relations.

Sets expressed in the language of the collection of granules formed by the fuzzy similarity relation. In this case these focal elements are fuzzy relations. The formulas describing the lower and upper approximations are given in the same way as already expressed above. The construct is visualized in Figure 6 where the fuzzy relations are portrayed as shadowed regions.

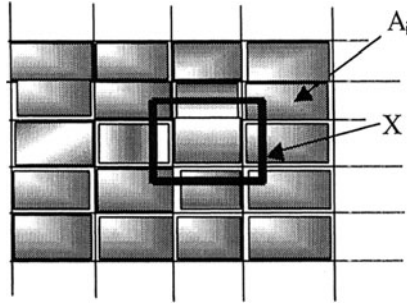


Figure 6. Set X expressed by means of fuzzy relations $\{A_i\}$.

Proceeding with the previous example, we have the fuzzy relations built over a family of fuzzy sets of speed such as $\{low, medium, high, very\ high\}$ and fuel consumption quantified through the collection of granules $\{low, medium, high\}$ while X is a Cartesian product of the interval of speed and fuel consumption, say $[50, 65] \times [10, 14]$.

Fuzzy set X expressed in the language of information granules being represented as fuzzy sets. In this case the formulas given above apply again; noticeably we compute degrees of contribution of the Cartesian products to the bounds. In this sense, the lower bound is associated with the possibility measure of X taken with respect to A_i . The upper bound comes in the form of the necessity measure.

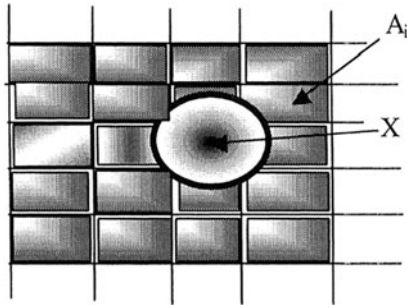


Figure 7. Fuzzy set X expressed by means of fuzzy relations $\{A_i\}$.

Coming back to the previous example, we have both the fuzzy relations built over a family of fuzzy sets of speed and fuel consumption X being is a Cartesian product of two fuzzy sets, say *low* speed and *high* fuel consumption.

Noticeably, in this combination of rough sets and fuzzy sets we clearly witness that they form a heterogeneous communication environment in which those rooted in the setting of fuzzy sets are expressed (articulated) in the language of rough sets and vice versa. In this sense, we may regard this fuzzy-rough (or rough-fuzzy)

combination as a communication framework for information granules expressed in different formal settings. This leads to a new view on the underlying concept of rough sets as the inherent vehicle of realization of exchange and interpretation of information (information granules) between two granular worlds (this idea is discussed in more detail in the future). At this point, we note that each of the cases illustrates how input granular information is “perceived” and interpreted in the language of the granules available in the given environment.

5. 5 SHADOWED SETS

The underlying motivation behind shadowed sets (Pedrycz, 1998; 1999) is the one about the localization of uncertainty of the membership grades and its “centralized” distribution across a fuzzy set.

Put it more formally, a shadowed set A defined in X is a granular construct realizing the mapping

$$A: X \rightarrow \{0, [0,1], 1\} \quad (8)$$

so for each element of X we assign either 0,1 or the entire unit interval. The condition $A(x) = 0$ states that x is excluded from the concept A , $A(x) = 1$ expresses that x fully belongs to A . A set of x 's for which this condition holds is called a *core* of the shadowed set. The third case where $A(x) = [0,1]$ quantifies a situation when *nothing* is known about the membership of the element (x) to A . This factor of ignorance becomes reflected by the interval type of membership, namely $[0,1]$. The set of arguments of X where this property holds is called a *shadow* of A . Conceptually, shadowed sets relate to the interval-valued fuzzy set as presented in Section 1. Likewise they also relate to rough sets as the shadows correspond to the boundaries of the rough set. Note, however, that this construct is not cast in the structure of some indiscernability relations as we encountered in rough sets.

Rather than that, the design of the shadowed set can be completed on a basis of some fuzzy set; we are referring to this construct as an induced shadowed set (being induced by some fuzzy set). Consider a unimodal fuzzy set A , see Figure 8. The design principle guiding the construction of a shadowed set concentrates on an idea of uncertainty localization. We reduce low membership grades to zero. High membership grades are elevated to 1. To maintain an overall balance of uncertainty we form a shadow that “absorbs” the uncertainty we have already eliminated by moving the membership grades up to one or reducing them down to zero. The total balance of uncertainty can be written down in the following form

$$W_1 + W_2 = W_3 \quad (11)$$

where W_1 is concerned with the region where the membership grades are reduced to zero; W_2 denotes the region over which we elevated the membership grades to one. W_3 balances this elimination of intermediate membership grades by forming a region over which we have interval-valued membership grades (in this case the entire unit interval) that is a shadow. Again, we stress that while a shadowed set can be introduced on its own, here, we approach this concept it from the standpoint of a construct being induced by some fuzzy set.

The essence of the construct is depicted in Figure 8. From the optimization standpoint, the realization of the above equality is accomplished by selecting a suitable value of α (threshold of the membership grades where the reduction or elevation of membership values occur). We assume that the problem is symmetric so that we elevate and suppress membership grades that are no lower than $1-\alpha$ and no higher than α . As a result, a choice of α is restricted to $(0, \frac{1}{2})$. Furthermore to emphasize that we are concerned with the non-numeric membership values we denote the resulting shadow by S , that is the shadowed set maps X into $\{0, S, 1\}$.

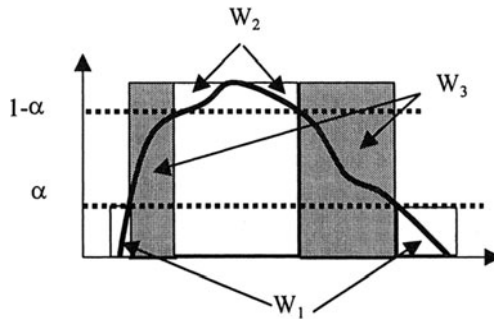


Figure 8. Shadowed set as a construct induced by fuzzy set: the design principle of uncertainty balancing.

Let us carry out detailed computations and discuss the results for some typical membership functions that is triangular, parabolic and Gaussian fuzzy sets.

Triangular membership functions

To simplify the problem, we fix the notation and consider a decreasing segment of the membership function (the same analysis is completed for the decreasing portion of the fuzzy set). The calculations of W_1 , W_2 , and W_3 lead to the following integrals, see Figure 9.

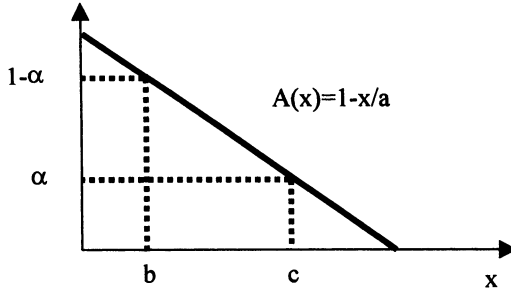


Figure 9. Determining a balance of uncertainty $W_1 + W_2 = W_3$ where $b = \alpha a$ and $c = a(1 - \alpha)$

$$W_1: \int_0^b (1 - A(x)) dx = \int_0^b (1 - 1 + x/a) dx = \frac{b^2}{2a} = \frac{\alpha^2 a}{2}$$

$$W_2: (c-b) = a(1-\alpha) - \alpha a = a(1-2\alpha)$$

and

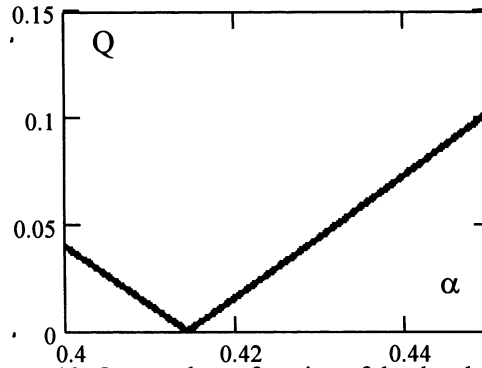
$$W_3: \int_c^a A(x) dx = \frac{\alpha^2 a}{2}$$

The final expression reads as

$$\alpha^2 a = a(1 - 2\alpha)$$

with the solution being equal to $\alpha = \sqrt{2} - 1 \approx 0.4142$.

The optimal value of α is the one for which the overall sum is equal to zero (meaning that the uncertainty balance has been accomplished). The plot of Q (that is $|W_1 + W_2 - W_3|$) versus α is shown in Figure 10.

Figure 10. Q treated as a function of the threshold α .

Parabolic membership functions

We are concerned with the membership of the form,

$$A(x) = \begin{cases} \left(1 - \frac{x}{a}\right)^2 & \text{for } 0 \leq x \leq a \\ 0 & \text{otherwise} \end{cases}$$

where $a > 0$. In the construction of the shadowed set, we confine ourselves to the decreasing portion of the membership function defined for the positive arguments of “x” (the optimization for the increasing portion of the membership is completed in an analogous fashion). The computations of the optimal threshold level involves three integrals whose boundaries are based on the cutoff points expressed as a function of α , namely

$$b(a, \alpha) = a(1 - (1 - \alpha)^{1/2})$$

and

$$c(a, \alpha) = a(1 - \alpha^{1/2}).$$

The integral over the region where the membership function is raised to 1 is equal to

$$\int_0^{b(a, \alpha)} (1 - A(x, a)) dx = a(2/3 - \sqrt{1 - \alpha} + 1/3(1 - \alpha)^{3/2})$$

The integral where we suppress the membership grades to zero is given as

$$\int_{c(a,\alpha)}^a A(x,a)dx = \frac{1}{3}\alpha^{3/2}a$$

Finally the region over which we form the shadow is equal to

$$W_3 = a(1-\alpha^{1/2}) - a(1-(1-\alpha)^{1/2})$$

The plot of Q treated as the absolute difference between W_1+W_2 and W_3 is shown in Figure 11.

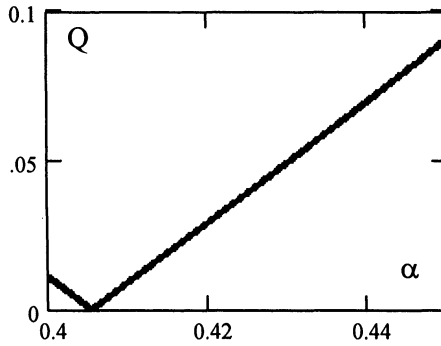


Figure 11. Plot of Q treated as a function of α .

The optimal value of α is equal to 0.405.

Gaussian membership functions

In this case we adopt numeric optimization, calculate the integrals numerically and then find an optimal value of α . The plot of Q is included in Figure 12 with the optimal value of α being equal to 0.394.

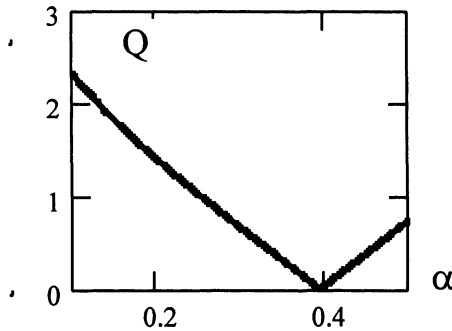


Figure 12. Q versus values of the threshold level ($\sigma=2$).

Operations on Shadowed Sets

The basic operations on shadowed sets are defined in the following tabular form

$A(x) \cap B(x)$	0	S	1
0	0	0	0
S	0	S	S
1	0	S	1

$A(x) \cup B(x)$	0	S	1
0	0	S	1
S	S	S	1
1	1	1	1

0	1
S	S
1	0
$A(x)$	$\overline{A(x)}$

where $S (= [0,1])$ denotes the shadow of the shadowed set. The results are intuitively appealing: for the *and* type of connective, we anticipate that when A is *unknown* then in spite of the numeric value of the membership function of B (equal to 1), the result comes as an interval. Similarly, when we consider a union of shadowed sets with A treated as unknown (that is represented as a shadow), then the result becomes a shadow unless the membership value of B is equal to 1.

Interestingly, shadowed sets are isomorphic with a three-valued logic as introduced by Lukasiewicz. In this logic having three truth values $\{0, \frac{1}{2}, 1\}$ we have the following matrices of the operations

$p(x) \text{ or } q(x)$	0	1/2	1
0	0	1/2	1
1/2	1/2	1/2	1
1	1	1	1

$p(x) \& q(x)$	0	1/2	1
0	0	0	0
1/2	0	1/2	1/2
1	0	1/2	1

0	1
1/2	1/2
1	0
$A(x)$	$\overline{A(x)}$

Hence the shadow corresponds to the intermediate truth value ($\frac{1}{2}$).

The concept of a shadowed set easily generalizes to shadowed relations. They can be defined as a part of the problem description or become induced by some fuzzy relation.

Transformations of Shadowed Sets

Shadowed sets defined in space X can be transformed to some other space Y by some function or relation. The underlying mechanism is based on the max-min composition that is the same as introduced in fuzzy sets. As we encounter the Boolean truth values (obviously we have shadows that are interval but from a computational standpoint they require processing 0s and 1s). The extension principle is again the computing mechanism being used in the transformation of X by “ f ”. We process the core of the shadowed set that leads to the core of the image of X , $f(X)$

$$\text{Core}(f(X))(y) = \sup_{x: y=f(x)} \{x \in \text{Core}(X)\} \quad (12)$$

Subsequently we compute the shadow of $f(X)$

$$\text{Shadow}(f(X))(y) = \sup_{x: y=f(x)} \{x \in \text{Shadow}(X)\} \quad (13)$$

For the transformation of X by a fuzzy relation we adopt a standard max-min composition. From the notation-based end, it is convenient to use the symbolic notation $\{0, S, 1\}$ as introduced when discussing the basic operations on shadowed sets. As the max-min composition is based on the max- and min operations only, the ensuing calculations become straightforward.

As an example, let us compute a transformation of $X = [0 \ S \ 1 \ S]$ by the Boolean relation

$$R = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Then the resulting shadowed set Y is equal to

$$Y(y_1) = \max\{\min(0,0), \min(S,1), \min(1,1), \min(S,0)\} = \max\{0, S, 1, 0\} = 1$$

$$Y(y_2) = \max\{\min(0,1), \min(S,0), \min(1,0), \min(S,0)\} = \max\{0,0,0,0\} = 0$$

$$Y(y_3) = \max\{\min(0,0), \min(S,1), \min(1,1), \min(S,0)\} = \max\{0, S, 1, 0\} = 1$$

$$Y(y_4) = \max\{\min(0,0), \min(S,0), \min(1,0), \min(S,1)\} = \max\{0, S, 0, S\} = S$$

so finally we get $Y = [1 \ 0 \ 1 \ S]$. Expanding this example, we now consider a shadowed relation

$$R = \begin{bmatrix} 0 & S & S & 0 \\ S & 1 & 1 & S \\ S & 1 & S & S \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Its composition with X leads to the shadowed set $Y = [S \ 1 \ S \ S]$.

5. 6 PROBABILISTIC SETS

The idea of probabilistic sets introduced by Hirota (Hirota, 1981) has been motivated by the orthogonality of the notion of granularity captured by fuzzy sets and randomness of the process of membership function estimation. This leads to the model in which for a given element of the universe of discourse X the grades of membership there are governed by some probability function (e.g., probability density function, pdf). In other words, $A(x, u)$ comes with a certain pdf (that can vary when moving between elements of X). We denote it by $p_x(u)$, $u \in [0,1]$ (as a matter of fact, the underlying pdf is defined in the unit interval). More descriptively, a probabilistic set A can be viewed as a random field that is a family of random variables indexed by " x ", $\{A_x\}$ where each A_x comes with its underlying pdf (p_x). A plot of some example probabilistic set is shown in Figure 13.

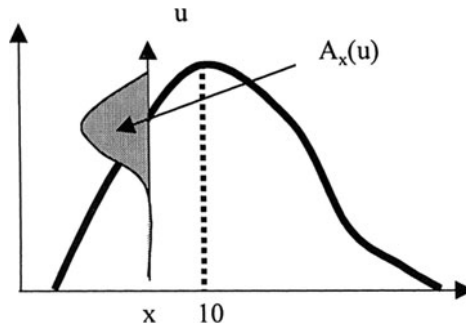


Figure 13. An example of a probabilistic set of *about* 10.

This model is appealing as it captures the nonuniqueness of the membership grades and attaches to each of them some useful probabilistic characteristics. If we admit a degenerate pdf where a total mass of probability is centered at a single membership value in $[0,1]$, namely

$$p_x(u) = \begin{cases} 1 & \text{if } u = u_0 \\ 1 & \text{otherwise} \end{cases}$$

then this definition returns a standard fuzzy set (which is an intuitively convincing observation). An important characterization of probabilistic sets comes in the form of moments where moments are computed in the same manner as encountered in the theory of probability. The r -th ($r = 0, 1, 2, \dots$) moment is defined in the following way

$$m_r = \int_0^1 u^r p_{A(x)}(u) du$$

Additionally, we introduce a family of centered moments

$$\mu_r = \int_0^1 (u - m_0)^r p_{A(x)}(u) du$$

It has been shown (Hirota, 1981) that the moments vanish quite quickly that is as the values of “ r ” increase, the values of m_r and μ_r tend to zero which is of practical relevance. Practically, one can concentrate on the zero moment (m_0) and the second centered moment (μ_2) as these two descriptors that convey a bulk of useful description of the probabilistic set. In this case, m_0 corresponds to the membership function of A . μ_2 is referred to as a vagueness of A which captures a phenomenon of granularity (spread) of the probabilistic set. If the vagueness is equal to zero and all higher moments vanish, then the probabilistic set returns a fuzzy set as its special case.

As the membership grades are governed by some probabilistic description, the operations on probabilistic sets such as union and intersection are subject to the mechanisms of processing of random variables. For two probabilistic sets A and B we compute their union and intersection (assuming the max and min operations of aggregation) in the following way (here we assume that X is finite),

- fix an element of X ; here we have $p_{A(x)}$ and $p_{B(x)}$ as the corresponding pdfs.
- Compute the pdf of the union of A and B in the form
- Compute the pdf of the intersection of A and B following the standard formula of probability calculus

Repeat the above computations for successive elements of X .

5. 7 INTUITIONISTIC FUZZY SETS

The idea of intuitionistic fuzzy sets is to directly capture the notions of membership and non-membership. As fuzzy sets concentrate on the membership facet (and that is really what membership grade captures), Intuitionistic fuzzy sets (Atanassov, 1986;

1994) can be regarded as an interesting generalization of fuzzy sets. As becomes obvious, both sets and fuzzy sets promote a concept of “positive” information about belongingness to a concept (set or fuzzy set) but do not address the issue of “negative” information that is quantify an extent to which an element is excluded from a given information granule. Formally, we develop the following structure

An intuitionistic set A defined in some space (universe) X is defined by two membership functions A_+ and A_- .

$$A = \langle A_-, A_+ \rangle \quad (14)$$

where $A_+(x)$ denotes a degree of membership and $A_-(x)$ stands for a degree of non-membership of $x \in X$ to A ; moreover we have $0 \leq A_-(x) + A_+(x) \leq 1$.

Intuitionistic sets are generalizations of fuzzy sets in the sense that we have $A_+(x) = A(x)$ and $A_-(x) = 1 - A(x)$ where obviously $A_+(x) + A_-(x) = 1$. The expression

$$\text{hes}(x) = 1 - A_+(x) - A_-(x)$$

is referred to as a hesitancy degree (hes) that quantifies our hesitation as to the given element x in A . Noticeably for any fuzzy set the hesitancy degree is equal to zero.

Intuitionistic fuzzy sets come with an interesting geometrical interpretation as shown in Figure 14. (cf. Atanasov, 1986). In this figure we show a distribution of possible values of $A_+(x)$ and $A_-(x)$ for a fixed element of X .

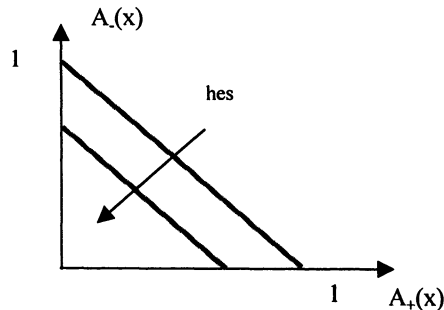


Figure 14. Geometric interpretation of intuitionistic fuzzy set with level of hesitancy

This figure shows that fuzzy sets are just specific cases of the above construct – A_+ and A_- are located on the line combining $(0, 1)$ and $(1, 0)$. Furthermore when move apart from this line, the level of hesitancy increases. To show the hesitancy, we can explicitly include its value in the graphic visualization by considering altogether

three dimensions of the intuitionistic fuzzy set that is A_+ , A_- and hesitancy, see Figure 15.

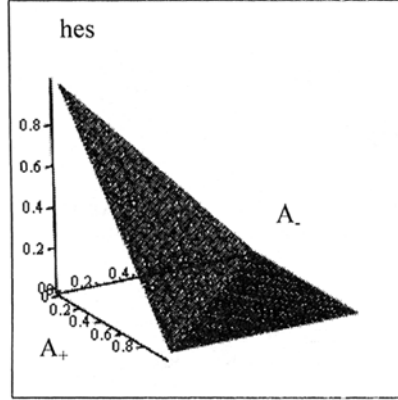


Figure 15. A 3D visualization of an intuitionistic set.

The operations on intuitionistic sets are expressed in a similar way as we encountered in fuzzy sets with a main difference concerning the formation of the degrees of membership and nonmembership.

The intersection of two intuitionistic fuzzy sets A and B is an intuitionistic fuzzy set with the membership function

$$\langle A_+(x) \text{ t } B_+(x), A_-(x) \text{ s } B_-(x) \rangle \quad (15)$$

where “t” is any t-norm and “s” denotes an s-norm (in general we do not require that these two operations are dual even though some specific generalizations of the intersection introduced by Atanassov (1986) imply this way of thinking). In virtue of this definition, we note that the degree of membership becomes reduced (as being an intuitively appealing phenomenon) $(A \cap B)_+(x) \leq \min(A_+(x), B_+(x))$ while the degree of non-membership is elevated, namely $(A \cap B)_-(x) \geq \min(A_-(x), B_-(x))$

For the union of two intuitionistic fuzzy sets we have the following definition

The union of two intuitionistic fuzzy sets A and B is an intuitionistic fuzzy sets with the membership functions

$$\langle A_+(x) \text{ s } B_+(x), A_-(x) \text{ t } B_-(x) \rangle \quad (16)$$

We say that A is included in B if for all $x \in X$ we get $A_+(x) \leq B_+(x)$ and $A_-(x) \geq B_-(x)$ that is we require that the degree of membership of A should be lower than the

degree of membership of B while the converse holds for the non-membership grades.

As an example, we consider the following two intuitionistic fuzzy sets A and B defined in the same finite space X

$$A = \{ [1.0 \ 0.6 \ 0.4 \ 0.5], [0.0 \ 0.2 \ 0.1 \ 0.0] \}$$

$$B = \{ [0.7 \ 0.2 \ 0.1 \ 0.4], [0.2 \ 0.6 \ 0.5 \ 0.6] \}$$

(the first vector A_+ collects the degrees of membership while the second, viz. A_- consists of the non-membership grades). If $t = \min$ and $s = \max$, we obtain

$$(A \cap B)_+ = \min(A_+(x), B_+(x)) = [0.7 \ 0.2 \ 0.1 \ 0.4]$$

$$(A \cap B)_- = \max(A_-(x), B_-(x)) = [0.2 \ 0.6 \ 0.5 \ 0.6]$$

so

$$(A \cap B) = \{ [0.7 \ 0.2 \ 0.1 \ 0.4], [0.2 \ 0.6 \ 0.5 \ 0.6] \}$$

For the sum of A and B we obtain

$$(A \cup B)_+ = \max(A_+(x), B_+(x)) = [1.0 \ 0.6 \ 0.4 \ 0.5]$$

$$(A \cup B)_- = \min(A_-(x), B_-(x)) = [0.0 \ 0.2 \ 0.1 \ 0.0]$$

For the product and probabilistic sum the resulting intuitionistic fuzzy set is

$$(A \cap B) = \{ [0.7 \ 0.12 \ 0.04 \ 0.2], [0.20 \ 0.68 \ 0.55 \ 0.60] \}$$

and

$$(A \cup B) = \{ [1.0 \ 0.68 \ 0.46 \ 0.70], [0.0 \ 0.12 \ 0.05 \ 0.00] \}$$

The linguistic hedges (modifiers) τ operating on the intuitionistic fuzzy set $\tau(A)$ give rise to the pair

$$\tau A^+(x) = A_+(x)^{m(\tau)} \quad \tau A_-(x) = 1 - (1 - A_-(x))^{m(\tau)}$$

where $m(\tau)$ is the exponent associated with the modifier (say *more or less* comes with $m(\tau) = 0.5$ while *very* concentrates the membership grade that is $m(\tau) = 2$). Using the same membership functions as studied above, *very* A implies the membership functions with the entries

$$\{ [1.00 \ 0.36 \ 0.16 \ 0.25], [0.00 \ 0.36 \ 0.19 \ 0.00] \}$$

while *more or less* A is defined as

$$\{[1.00 \ 0.77 \ 0.63 \ 0.71], [0.00 \ 0.11 \ 0.05 \ 0.00]\}$$

For the two intuitionistic fuzzy sets we compute the distance between them by considering the distance between the membership, non-membership degrees and the resulting degrees of hesitancy (Szmidt and Kacprzyk, 2000). E.g., the Hamming distance would then read as follows

$$d_H(A, B) = \sum_{i=1}^n (|A_+(x_i) - B_+(x_i)| + |A_-(x_i) - B_-(x_i)| + |\text{hes}_A(x_i) - \text{hes}_B(x_i)|)$$

(in the above expression we assumed that the universe of discourse has a finite cardinality being equal to n); evidently $d_H(A, B) \in [0, n]$. (the other distances such as the Euclidean, Tschebyschev, etc. are introduced in the same manner).

One could mention that another definition of the distance can be introduced where we are looking only at the distance between the degrees of membership and non-membership.

5. 8 PROBABILITY OF GRANULAR CONSTRUCTS: GRANULARITY AND THEIR EXPERIMENTAL RELEVANCE

A number of observations made so far emphasizes that information granules represented by sets, fuzzy sets, rough sets, etc. and probability are orthogonal meaning that their characteristics are distributed along two orthogonal axes and they supplement each other. This stems from a fundamental fact that probability is concerned about uncertainty as to occurrence or non-occurrence of some phenomenon. The other mechanisms of information granulation are concerned about an extent to which some element becomes a part of the concept.

The orthogonality of the two constructs (viz. granular information and probability) implies that they supplement each other. As a matter of fact, we can assess any information granule $G \in \mathcal{G}(X)$ with regard to its granularity and its experimental evidence conveyed by the probabilistic characteristics of such granule. We start with a discussion of the probability of fuzzy events.

granularity and probability of information granules. Information granules can be described in terms of their sizes (that is granularity) and probability. We require that the information granule is *specific* enough so that its granularity is high (so the

cardinality $\text{Card}(X)$ becomes low enough and does not exceed some threshold level) while at the same time they carry enough *experimental evidence*. This requirement is quantified in terms of the probability of the event that should be higher than some predefined threshold value β . Formally, we can express this requirement as, see Figure 16,

$$\text{Card}(X) \leq \alpha \quad \text{and} \quad \text{Prob}(X) \geq \beta \quad (17)$$

where $\alpha > 0$ and $\beta \in [0,1]$.

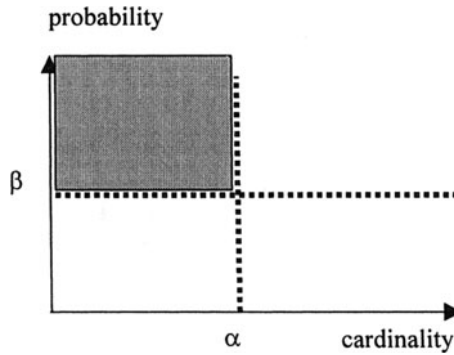


Figure 16. Information granules of acceptable specificity and experimental evidence (shown as a shadowed area).

Several of situations summarized in Figure 17 illustrates how fuzzy probabilities are determined in case of some fixed Gaussian probability density function and Gaussian fuzzy sets with different modal values and spreads.

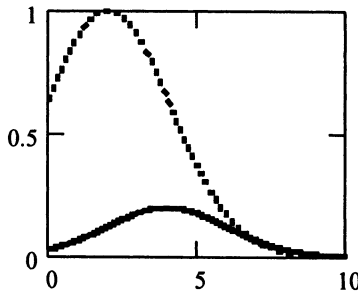


Figure 17(a). Gaussian fuzzy sets (dotted line) and probability density function (solid line) - selected examples: $P(A)=0.575$, $m=2$; $\sigma=3$

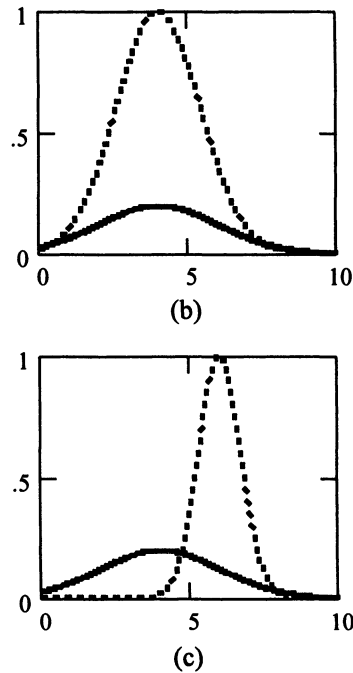


Figure 17. Gaussian fuzzy sets (dotted line) and probability density function (solid line) - selected examples: (b) $P(A)=0.577$, $m=4$, $\sigma=2$, (c) $P(A) = 0.214$, $m=6$, $\sigma=1$.

Apparently when A coincides with the highest values of the probability density function (as shown in Figure 17(b)), it becomes more specific and still assures a high level of experimental relevance.

The plot of the probability of the fuzzy event for some changes in the parameters of A (modal value and its spread) is portrayed in Figure 18. As expected, with the more visible departure of the modal value from the mean value of the pdf, the probability of the fuzzy event gets lower. The increase in the spread of the fuzzy set helps offset this decrease; its higher values increase the probability of the fuzzy event.

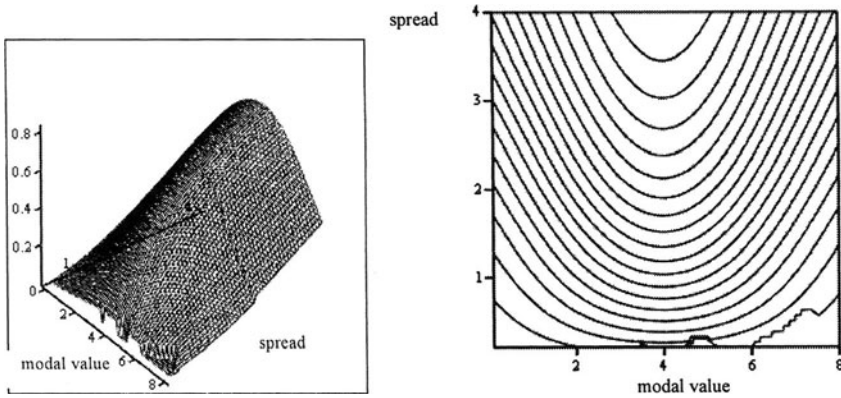


Figure 18. $P(A)$ as a function of its parameters: modal value and its spread.

Set-based information granules In case of set-based information granules, we end up with a standard definition of an event encountered in probability theory,

$$\text{Prob}(A) = \int_{x:A(x)=1} p(x)dx = \int_{\mathbf{x}} A(x)p(x)dx$$

where A is a set. For an interval $A=[a, b]$, we end up with the following integral

$$\text{Prob}(A) = \int_a^b p(x)dx$$

Interval-valued fuzzy sets For interval-valued fuzzy sets, we end up with the bounds on the probabilities related directly with the bounds of the intervals in the membership grades A_1 and A_2 , namely

$$\text{Prob}(<A_1, A_2>) = \left[\int_{\mathbf{x}} A_1(x)p(x)dx, \int_{\mathbf{x}} A_2(x)p(x)dx \right]$$

Shadowed fuzzy sets For shadowed sets, where we encounter an inherent uncertainty in the membership grades (shadows), we can compute the bounds of the membership. Let the shadowed set be described by the parameters (a, m, n, b) with the shadows distributed over $[a, m]$ and $[n, b]$ while the core is expressed as $[m, n]$. The lower bound of probability of the shadowed set is computed as

$$\text{Prob} = \int_m^n p(x) dx$$

while the upper bound comes in the form

$$\text{Prob} = \int_m^n p(x) dx + \int_a^m p(x) dx + \int_n^b p(x) dx = \int_a^b p(x) dx$$

The larger the shadows, the more distinct the bounds of the probability of the event of the information granule.

Similarly, for a rough set we compute the bounds of the probability by taking its lower and upper approximation.

5. 9 CONCLUDING COMMENTS

Granularity of information exhibits a number of various facets. We showed a series of generalizations that capture this aspect. Most of them generalize along the line of granular membership values so that they are conveniently described in the format

$$X: X \rightarrow \mathcal{G}[0,1]$$

where $\mathcal{G}[0,1]$ denotes some granular construct distributed over the unit interval). More specifically we have $\mathcal{H}[0,1]$ that leads to interval-valued fuzzy set, $\mathcal{A}[0,1]$ with the type-2 fuzzy sets, three-valued mixed set $\{0, 1, S\}$ that occurs in case of shadowed sets, to enumerate the main constructs discussed in this chapter.

The orthogonality of information granules and probability is another important feature of granular computing. We have emphasized that these two concepts capture two different aspects of information. This helps us establish a general criterion of acceptability or relevance of information granules viewed from the standpoint of minimal specificity (that imposes a certain constraint on their size) and a minimal experimental evidence (which makes them relevant from the standpoint of numeric data available in the problem at hand).

REFERENCES

- Atanassov, K. (1986), Intuitionistic fuzzy sets, *Fuzzy Sets and Systems*, **20**, 87-96.
 Atanassov, K. (1994), New operations defined over the intuitionistic fuzzy sets, *Fuzzy Sets and Systems*, **61**, 137-142.

Chen, S.M., Hsiao, W.H., Jong, W.T. (1997), Bidirectional approximate reasoning based on interval-valued fuzzy sets, *Fuzzy Sets and Systems*, **91**, 339-353.

Dubois, D., Prade, H.(1992), Rough fuzzy sets and fuzzy rough sets, *Int. J. General Systems*, **17**, 203-232.

Gorzalczany, M.B. (1987), A method of inference in approximate reasoning based on interval-valued fuzzy sets, *Fuzzy Sets and Systems*, **21**, 1-17.

Hirota, H. (1981), Concepts of probabilistic sets, *Fuzzy Sets & Systems*, **5**, 31-46.

Karnik, N.N., Mendel, J.M. (2001), Operations on type-2 fuzzy sets, *Fuzzy Sets and Systems*, **12**, 327-348.

Mendel, J.M., John, R.I.B. (2002), Type-2 fuzzy sets made simple, *IEEE Trans. on Fuzzy Systems*, **10**(2), 117-127.

Pawlak, Z. (1982), Rough sets, *Int. J. Inform. Comp. Sci.*, **11**(5), 341-356.

Pedrycz, W. (1998), Shadowed sets: representing and processing fuzzy sets, *IEEE Trans. on Systems, Man, and Cybernetics*, part B, **28**, 103-109.

Pedrycz, W. (1999), Shadowed sets: bridging fuzzy and rough sets, In: Pal, S. K., Skowron A.(eds.), *Rough Fuzzy Hybridization. A New Trend in Decision-Making*, Springer Verlag, Singapore, pp. 179-199.

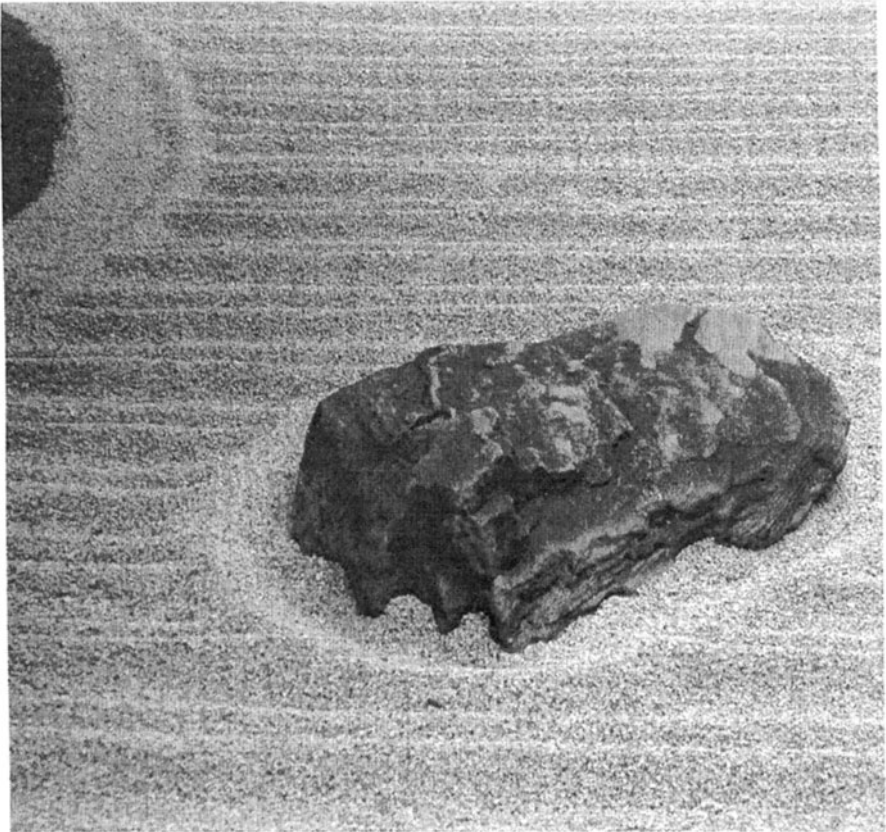
Szmidt, E., Kacprzyk, J. (2000), Distances between intuitionistic fuzzy sets, *Fuzzy Sets and Systems*, **114**, 505-518

Zadeh, L.A. (1968), Probability of fuzzy events, *J. Math. Anal and Appl.*, **22**, 421-427.

Zimmermann, H.J. (2001), *Fuzzy Set Theory and Its Applications*, 4th Edition, Kluwer Academic Publishers, Boston, Dordercht.

PART II

ALGORITHMS OF INFORMATION GRANULATION



FROM NUMBERS TO INFORMATION GRANULES

We start Part II of this book by exploring the idea of granular clustering as a way of finding a structure in heterogeneous data. The main features of the proposed approach are: (a) a noninvasive exploration of data carried out under weak assumptions made as to the nature of the data, (b) transparency of the constructed information granules assuming a form of hyperboxes in the data space. We introduce a compatibility measure that helps express a degree of “similarity” between two information granules and takes into account a distance between the granules as well as their size. We show how to “grow” clusters through a process of merging existing data points that exhibit high values of the *compatibility measure*. The clustering algorithm is discussed along with a comprehensive validation mechanism of the resulting structures (a collection of information granules). We formulate a problem of feature analysis in the setting of the information granules and introduce some quantitative measures describing each feature. Numerical experiments use two-dimensional synthetic data as well as multi-dimensional Boston data available on the WWW.

6. 1 INTRODUCTORY COMMENTS

The progress of scientific and engineering achievement was for a long time equated to the improved accuracy of measurements and system models. However, the vast amount of accurate numerical data that became available over the last two decades, in all domains of human activity, highlighted the fact that there is an urgent need for a more human-like processing of information, namely automated generalization and abstraction. This was first pointed out in the pioneering work of Zadeh (1979) who coined the term of *information granulation* and emphasized the fact that the plethora of detail does not amount to knowledge. In many ways this new realization has been recognized implicitly throughout the history of the development of science, since the ability to abstract from detail and to generalize the conclusions was always seen as one of the indicators of human intelligence. However, in the changed circumstances of having vast volumes of data that preclude human analysis, the importance of Zadeh’s contribution derives from his suggestion that there are basic mechanisms

that underlie this process of *knowledge building*. In other words, the suggestion is that the process of knowledge building can be automated. The discovery of these mechanisms is a key challenge of Granular Computing (GC). This chapter introduces the basic framework that facilitates *transition from numbers to semantically richer information granules* and it sets the scene for the information granulation algorithms discussed in subsequent chapters.

The mathematical formalism of the interval analysis provides a robust framework for the analysis of information density of the granular structures that emerge in the process of granular clustering. This reflects the intuitive objective of matching the granularity of data items used to describe the physical systems to the structure of these systems. In this sense the granulation process introduced in this chapter attempts to achieve the highest possible generalization while maintaining the specificity of data structures.

6. 2 INFORMATION GRANULES AND INFORMATION GRANULATION

Most experimental data available in a raw form are numeric. Granulation of information happens through a process of data organization and data comprehension. Interestingly; humans granulate information almost in a subconscious manner. This eventually makes the ensuing cognitive processes so effective and far superior over processes occurring under the auspices of machine intelligence. Two representative categories of problems in which information granulation emerges in a profound way involve processing of one and two-dimensional signals. The first case, we are concerned primarily with temporal signals. The latter case pertains to image processing and image analysis. In signal processing, analysis and interpretation the granules arise as a result of temporal sampling and aggregation. Several samples in the same time window can be represented as an information granule. In the simplest case, such interval can be formed by taking a minimal and maximal value of the signal occurring in this window of granulation, see Figure 1.

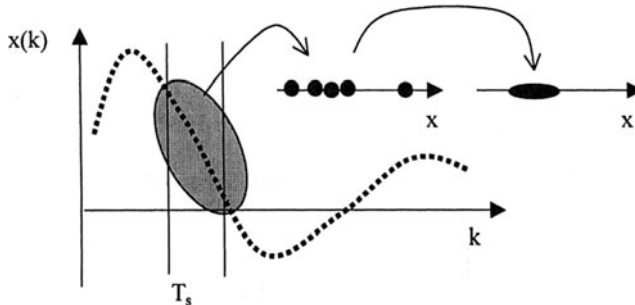


Figure 1. A fragment of a time series and its granulation through sampling (T_s denotes a sampling interval).

Information granulation has been studied in (Bargiela, 2001; Bargiela, Pedrycz, 2002; Pedrycz, 1997; Pedrycz, 2001) both in terms of the concept itself, computational aspects of it as well as resulting structures.

In the overall presentation we adhere to the notation introduced in Chapter 2. A hyperbox $[b]$ defined in \mathbf{R}^n is fully described by its lower (b^-) and upper corner (b^+), where b^- and b^+ are vectors in \mathbf{R}^n . Using b^- and b^+ we can express the hyperbox as $[b]=[b^-, b^+]$. An evident ordering relationship between hyperbox corners holds true, that is $b^- \leq b^+$. If $b^- = b^+$ then the hyperbox reduces to a single point (numeric datum). The volume of $[b]$, denoted by $\text{volume}([b])$, is viewed as a measure of specificity of the information granule. The point-sized hyperbox comes with the highest specificity that becomes reduced once the volume increases. Computationally, it is advantageous to consider the expression $\exp(-\text{volume}(.))$ which captures the same aspect of granularity yet this measure is normalized as it attains 1 for the numeric datum and reduces to zero once the hyperbox starts growing.

It is instructive to elaborate on the use of such language of information granules in the realm of PR (in which we are quite commonly confined to the language of probability and probabilistic granules articulated as probability functions or probability density functions). The transparency of the results is a key factor. To make this point evident, refer to Figure 2 illustrating two granular constructs. In the first case, a two-dimensional box captures the essence of the data: we may state that the Cartesian product of $[a,b]$ and $[c,d]$, $[x] = [a,b] \times [c,d]$ “covers” the data. Moreover, both features (intervals) maintain their identity. In contrast, the language of ellipsoidal information granules (that can be quite expressive) does not come with the same transparency as hyperboxes, Figure 2(b). Obviously one can project the ellipsoid on the corresponding features. Note however that the reconstruction being treated as a Cartesian product $[y] = [e, f] \times [g, h]$ could be quite different from the original granule $\Omega \neq [y]$.

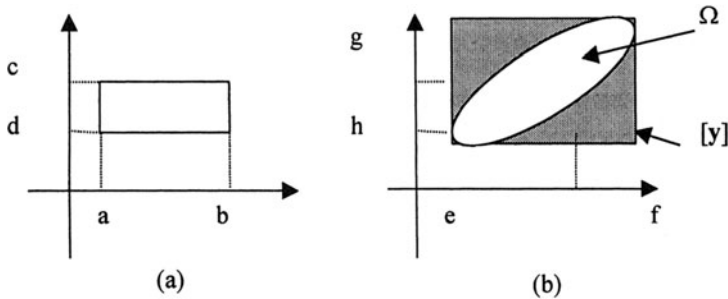


Figure 2. Expressing data in the language of hyperboxes (a) and ellipsoidal information granules (b); note a reconstruction deficiency caused by the dependency between the features.

6. 3 THE PRINCIPLE OF GRANULAR CLUSTERING

Before we proceed with the details of the clustering technique for granular data, it is instructive to discuss the underlying principle, learn how the process proceeds and concentrate on the interpretation of some results generated by the proposed clustering mechanism.

As emphasized in the literature (Andenberg, 1973; Bezdek, 1981), the essence of clustering (unsupervised learning) is to discover a structure in data. In essence, almost all existing clustering techniques operate on numeric objects (vectors in \mathbf{R}^n) and produce representatives (say, prototypes) that are again entirely numeric. In this sense, their form does not reflect how much data points they represent and how the distribution of these data points looks like (obviously, the nature of data is captured by a pertinent allocation of the prototypes). In the design of the clustering method, we add an extra dimension of granularity that helps sense the structure in the data as it becomes unveiled during the formation of the clusters.

Conceptual Design

This approach introduced here is very much different in many ways from the others. The leitmotiv is the following:

An abstraction (no matter whether dealing with numeric or granular elements) is achieved through *condensation* of original data elements into granules whose location and granularity reflects the essence of the structure of data. The more condensation, the larger the sizes of the information granules that realize this aggregation.

At the algorithmic end, the granular clustering is carried out as the following iterative process

- find the two closest information granules (where the idea of compatibility guiding this search of information will be quantified later on) and on this basis build a new granule embracing them. In this way, one condenses the data while reducing the size of the data set
- repeat the first step until enough data condensation has been accomplished (here one has to come up with a certain termination criterion or introduce a sound validation mechanism)

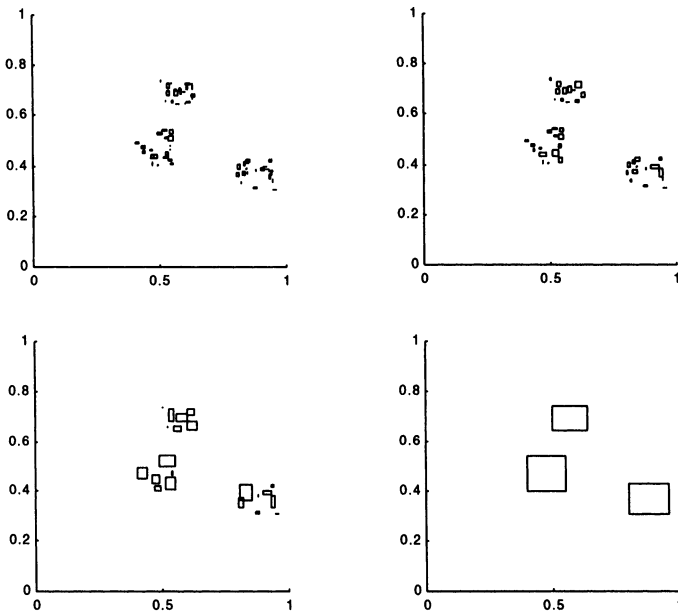


Figure 3. Several snapshots of cluster growing over the clustering process; observe that small information granules forming at the initial stage (first iteration) that are grouped in some well-confined regions and give rise to three apparent large information granules at the later stage of clustering.

Figure 3 illustrates how the granular clustering works. We start from a collection of small information granules (these are original data) and start growing larger information granules. Noticeably, through their growth they tend reflecting the essential characteristics of the original data. The size of the granules reflects quite evidently how much they have incorporated the original data and convey an extra message about their dispersion (distribution).

Considering a way in which the data points are merged together this approach resembles techniques of aggregative hierarchical clustering. There is a striking difference though: in hierarchical clustering we deal with numeric objects and the clusters are sets of the same objects. No conceptually new entities are formed. Here, we “grow” the clusters: from iteration to iteration they tend to form larger hyperboxes. Moreover the nature of these hyperboxes help monitor the clustering process more thoroughly and raise awareness about terminating the clustering. Essentially, once we have found that the evolved boxes become distant in the state space, the process of clustering (forming combined boxes) is terminated.

By the same token, this concept should be contrasted with the idea of min-max clustering discussed by Simpson (Simpson, 1992; Simpson, 1993) as this technique seems to bear some resemblance with the method presented here. The similarity is superficial though. First, the Simpson's method deals only with point-size data while the data considered here is a mix of points and hyperboxes in the pattern space. Second, the fuzzy membership functions of the information granules (clusters) as proposed by Simpson promote formation of clusters that are having largely varying sizes in various dimensions which is exactly the opposite to what we are trying to promote through the "compatibility measure" (discussed in Section 4). To make this point clear we present in Figure 4 a representative of a class of membership functions proposed by Simpson and refer the reader to Figure 10 for comparison with the functions that have been utilized in the granular clustering algorithm.

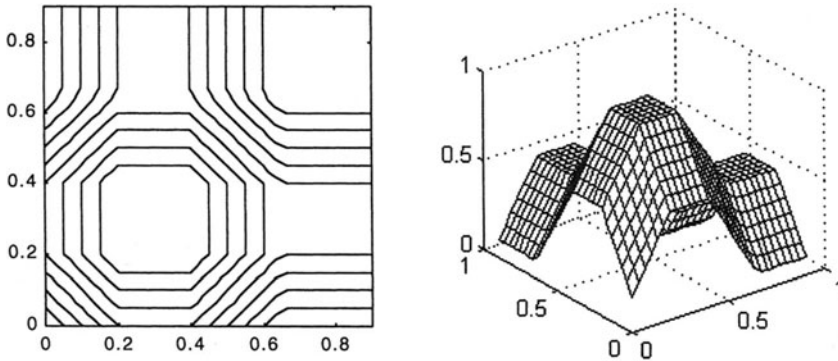


Figure 4. Simpson's membership function (as presented in [15]) for the hyperbox defined by the min point $\mathbf{x}^-=(0.2, 0.2)$ and max point $\mathbf{x}^+=(0.4, 0.4)$. Sensitivity parameter γ is equal to 4.

Interpretation and Validation of Granular Clustering

Clustering comes with a significant number of cluster validity indexes whose role is to identify the most "plausible" number of clusters. They help navigate the clustering process by stating what number of clusters should be. Commonly, their behavior does not lead to clear conclusions. What could be even worse, they may generate conflicting suggestions as to the termination condition (that is the number of clusters). In granular clustering we take another position. As the clusters capture the core of the data (and obviously, this is regarded as an important benefit of the method), our conjecture is that such core should help establish a sound platform of assessment of the structure (granular clusters).

When progressing with an expansion of the information granules, a certain criterion worth investigating deals with measuring a volume of the smallest granule (V_{\min}) that is constructed at this particular step (more specifically, we determine $e^{-V_{\min}}$; the details will be covered in Section 4.1). The main point is that if such minimal volume grows quickly to cluster two granules, then it can be deduced that the compatibility of the component granules is low and the clustering process can be completed.

Again, it is worth emphasizing that the granularity of data adds an extra important dimension to any processing. Not only a location of the information granule is essential but also its size plays a crucial role in the process of clustering and afterwards during the validation of the clusters.

6. 4 THE COMPUTATIONAL ASPECTS OF GRANULAR COMPUTING

There are two essential functional elements of granular clustering that need to be constructed prior to moving to the detailed computing. These concern a way in which a distance between two information granules is determined and how we compute an inclusion relation between them. While the definitions generalize to a multidimensional case, we focus here on a two-dimensional case. Note also that these two concepts work for heterogeneous data, i.e. granules and numeric entities.

Defining Compatibility Between Information Granules

In this section, we discuss details in which a compatibility and inclusion between two information granules are computed. The issue is more complicated than in a numeric case as these notions are granular and therefore the definitions of compatibility and inclusion should reflect this aspect as well.

Consider two information granules (hyperboxes) $[a]$ and $[b]$. More explicitly, we follow a full notation $[a]=[a^-, a^+]$ and $[b]=[b^-, b^+]$ to point at their location in the space. The expression of compatibility, $\text{compat}([a], [b])$ involves two components that is a distance between $[a]$ and $[b]$, $d([a],[b])$, and a size of a newly formed information granule that comes when merging $[a]$ and $[b]$. The distance $d([a],[b])$ is defined on a basis of the distance between its extreme vertices, that is

$$d([a],[b]) = (\|a^- - b^-\| + \|a^+ - b^+\|)/2 \quad (1)$$

that is an average of the two distances. Obviously $\|\cdot\|$ is a distance defined between the two numeric vectors. To make the framework general enough, we treat $\|\cdot\|$ as an L_p distance, $p \geq 1$. By changing the value of “p” we sweep across a spectrum of well known distances that depend upon a particular value of “p”. For instance, $p = 1$

yields a Hamming distance, L_1 . The value $p = 2$ produces a well – known Euclidean distance, L_2 . For $p = \infty$ we refer to a Tchebyshev distance, L_∞ .

Once $[a]$ and $[b]$ have been combined giving rise to a new information granule $[c]$, its granularity can be captured by a volume, $V([c]) = \text{volume}([c])$ computed in a standard fashion

$$V([c]) = \prod_{i=1}^n \text{length}_i([c]) \quad (2)$$

where

$$\text{length}_i([c]) = \max(a_i^+, b_i^+) - \min(a_i^-, b_i^-) \quad (3)$$

$i=1, 2, \dots, n$. For details, refer to Figure 5.

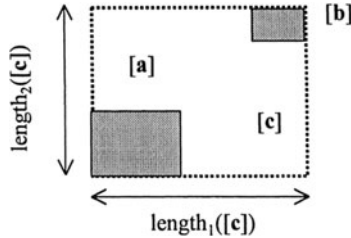


Figure 5. Information granule $[c]$ as a result of combining $[a]$ and $[b]$.

The two expressions (1)-(2) are the contributing factors to the compatibility measure, $\text{compat}([a],[b])$ to be defined now in the form

$$\text{compat}([a],[b]) = 1 - d([a],[b])e^{-\alpha V([c])} \quad (4)$$

The rationale behind the above form of the compatibility measure is as follows. In clustering we aggregate (cluster) two information granules that are the closest viz. their compatibility measure is the highest, $\text{compat}([a],[b]) = 1$. In light of the above criterion, the candidate granules to be clustered should not only be “close” enough (which is reflected by the distance component) but the resulting granule should be “compact” (meaning that the size of the granule in every dimension is approximately equal). The second requirement favors such $[a]$ and $[b]$ that give rise to a maximum volume for a given $d([a],[b])$, in other words it stipulates formation of hyperboxes that are as similar to hypercubes as possible. The particular exponential form of this expression has to do with the normalization criterion so that all values are kept in the unit interval. In particular, the volume of a point produces $e^0 = 1$ While the volume increases, its exponential function goes down to zero. The parameter α balances the

two concerns in the compatibility measure and is chosen so as to control an extent to which the volume impacts the compatibility measure.

The compactness factor ($e^{-\alpha V(\ell)}$) introduced in the compatibility measure is critical to the overall processing (viz. clustering) of the information granules. By contrast, it is not essential and does not play any role when we proceed in a standard way and do not attempt to develop granules but retain a cluster of numeric data. To retain the values of the compatibility measure to the unit interval, we confine the data to the unit hypercube $[0,1]^n \subset \mathbf{R}^n$ (in other words we normalize the data before computing the value of (4) and consider a normalized version distance assuming the values in the unit interval).

To gain a better insight of what really is accomplished when using the above compatibility measure, let us study two points (numeric values) \mathbf{a} and \mathbf{b} situated in \mathbf{R}^2 . Furthermore let \mathbf{a} be fixed and located at the origin of the coordinates while we allow \mathbf{b} with some flexibility. The distance $d(\mathbf{a}, \mathbf{b})$ is just a standard Euclidean distance. It becomes obvious that all elements (\mathbf{b} s) located on a circle of a fixed radius exhibit the same distance value. Restrict now a choice of \mathbf{b} s from this pool. If we connect \mathbf{a} and any of such \mathbf{b} s, the resulting volume changes its value depending upon the location of \mathbf{b} . Interestingly, out of all \mathbf{b} s, there are four of them (located on this circle) for which the volume of the resulting attains its maximum. This happens if such box (viz. the information granule formed by clustering \mathbf{a} and \mathbf{b}) is a square, refer again to Figure 6. In other words, the compatibility measure attains a maximal value there.

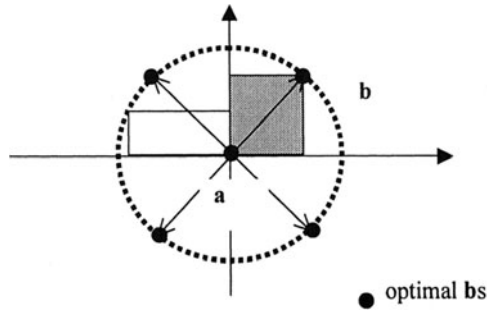


Figure 6. The calculations of the compatibility measure; note that there are four possible candidates (\mathbf{b} s) on the circle that maximize this measure.

If we plot the compatibility measure as a function of τ (where τ is an angular position of \mathbf{b}), we can easily see that the values of the compatibility measure are modulated by the angle (or equivalently the shape of the resulting information granule (hyperbox) $[\mathbf{c}]$), see Figure 7.

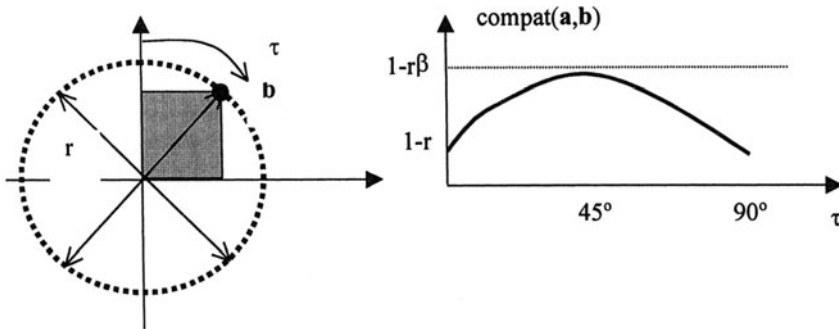


Figure 7. The compatibility measure expressed as a function of τ (the plot here plot is restricted to the first 90° degrees); $\beta = e^{-\alpha r / r^2}$.

More importantly, the above graphical considerations shed light on the geometry of the information granules that are preferred by the introduced compatibility measure. Such preference reflects a principle that may be coined as a *principle of balanced information granularity*. In a nutshell, in building new information granules, we prefer to have entities whose granularity is balanced along all dimensions (variables) rather than constructing granules that are highly unbalanced. A number of selected examples of varying granularity are portrayed in Figure 8.

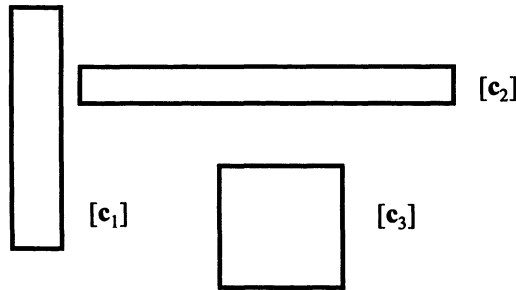


Figure 8. Examples of information granules characterized by various degrees of balance of information granularity; note that $[c_1]$ and $[c_2]$ are highly unbalanced as exhibiting different levels of information specificity along one of the variables ($[c_1]$ and $[c_2]$ with high specificity along x_1 and x_2 , respectively) while $[c_3]$ is well-balanced.

When changing the distance function to the Hamming ($p = 1$) and Tchebyshev distance ($p = \infty$), and carrying out the calculations of the compatibility measure, see Figure 6, now we have a number of Bs to choose from yet this selection can be made from different geometrical figures (that is a diamond and a square), Figure 9.

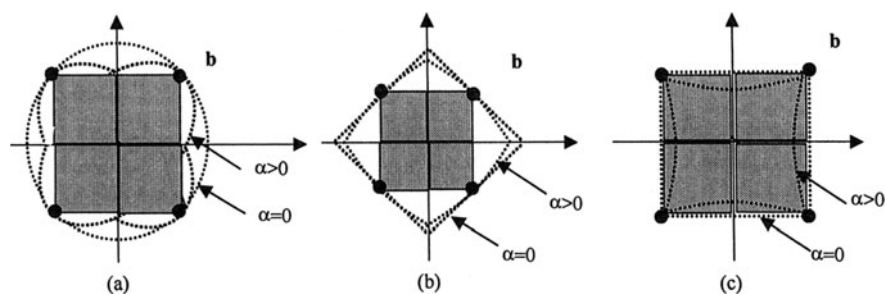
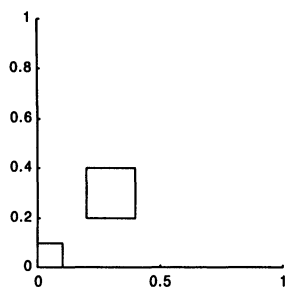
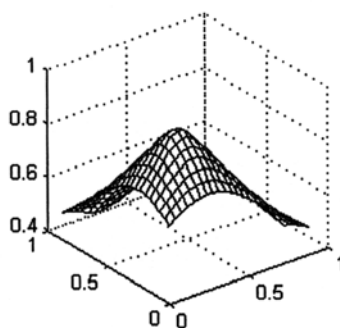
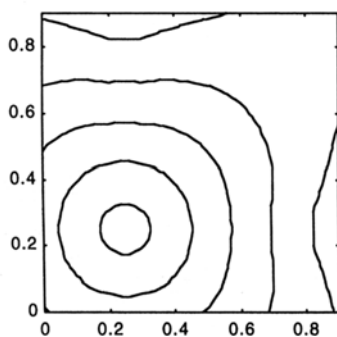


Figure 9. Identification of b s leading to the highest value of the compatibility measure.

Moving on to the case where $[a]$ and $[b]$ are information granules (hyperboxes), the resulting plots visualizing the compatibility measure are collected in Figure 10.



(a) Two hyperboxes representing information granules in a unit box in \mathbb{R}^2



(b) Compatibility measure with L_2 distance measure

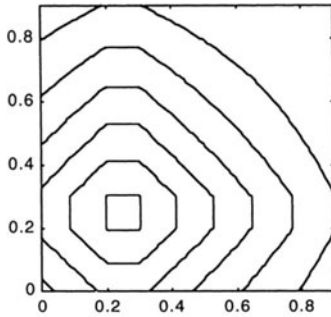
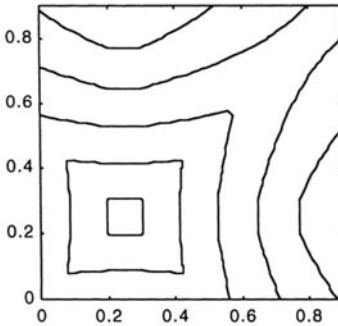
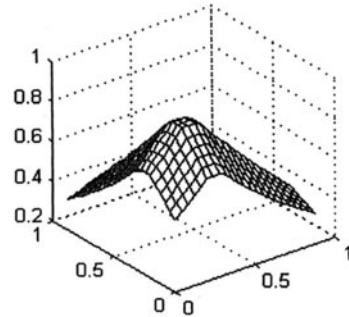
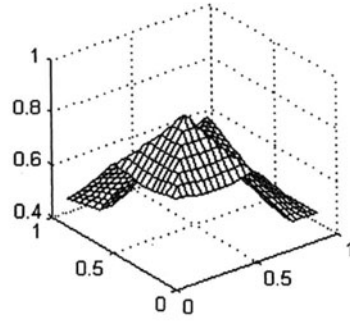
(c) Compatibility measure with L_1 distance measure(d) Compatibility measure with L_∞ distance measure

Figure 10. Comparison of compatibility measures obtained with various distance measures. Note the preference that the compatibility measure gives to hyperboxes that are well balanced in all dimensions. This contrasts with the membership function proposed in [Simpson, 1993] and illustrated in Figure 4.

As the clustering proceeds (refer to Figure 3) the process of merging the progressively less closely associated patterns finds its reflection in the gradual reduction of the compatibility measure (4). A typical plot of the evolution of the compatibility measure over the complete clustering cycle is shown in Figure 11.

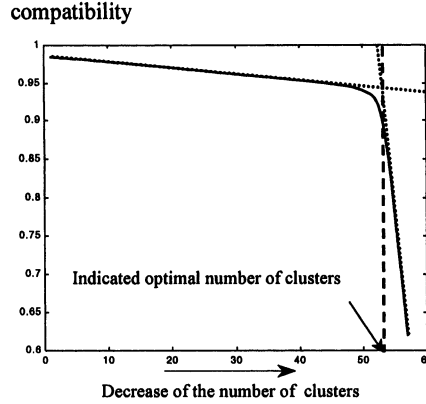


Figure 11. An example of the evolution of the compatibility measure over the full cycle of the clustering process.

It is self evident that the proximity of patterns that are being merged into granules at the early stages of the clustering process, is reflected in the relatively small gradient of the compatibility measure curve. By contrast, a large gradient of the curve, at the final stages of the clustering, indicates merging of highly *incompatible* clusters. The compatibility measure curve provides therefore a convenient reference for identifying which number of clusters captures the essential characteristics of the input data while providing the best generalization of them. The intersection of the two gradient lines (as visualized in Figure 11) can be used as an approximation to the optimal number of clusters. This number provides a good starting point in the subsequent optimization of the overlap of the identified clusters as discussed below.

Referring to the compatibility index, we can consider its modified form where we consider a sum of the sides(edges) of the hyperboxes that is

$$\text{compat}([a],[b]) = 1 - d([a],[b])e^{-\alpha L([c])} \quad (5)$$

with

$$L([c]) = \sum_{i=1}^n \text{length}_i([c]) \quad (6)$$

(considering the nature of these indexes, we refer to the first index as volume-driven while the second is edge-driven).

To compare these two forms of the compatibility index, we consider a simple two-dimensional case in which both a and x are numeric. We allow x to move on a unit circle while a is located at the origin of the coordinates, see Figure 12.

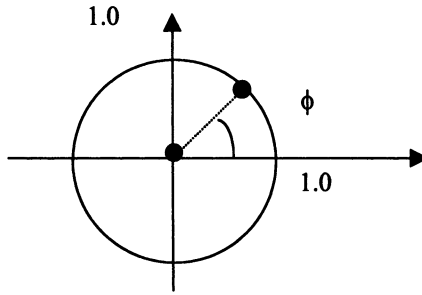


Figure 12. Computing compatibility measure for \mathbf{a} and \mathbf{x} and expressed as function of ϕ .

In this way the distance is always equal to 1 and the compatibility can be expressed by a single angle ϕ , namely

- for the volume driven version

$$\text{compat}(\mathbf{a}, \mathbf{x}) = 1 - e^{-(\sin \phi \cos \phi)}$$

- for the edge-driven version

$$\text{compat}(\mathbf{a}, \mathbf{x}) = 1 - e^{-(\sin \phi + \cos \phi)}$$

The plots of the compatibility measures are shown in Figure 13. It becomes obvious that the highest compatibility value is achieved for the same value of the angle, i.e., $\phi = \pi/4$.

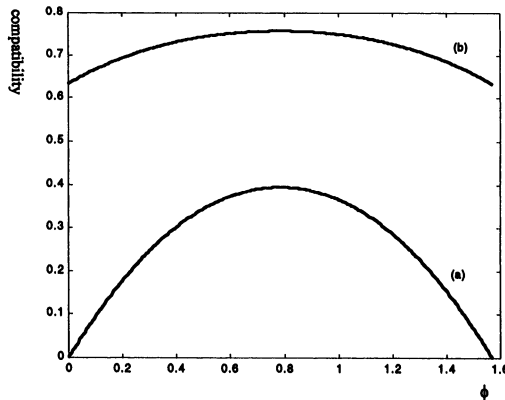


Figure 13. Compatibility measures as a function of ϕ : (a) volume oriented, (b) edge-oriented; ϕ is constrained to $[0, \pi/2]$.

These compatibility measures exhibit a visible difference when we look at their sensitivity defined as

$$\text{sens}(\phi) = \left| \frac{\partial \text{compat}(\mathbf{a}, \mathbf{x})}{\partial \phi} \right|$$

Figure 14 reveals that the compatibility based on the volume of the information granule comes with a higher sensitivity.

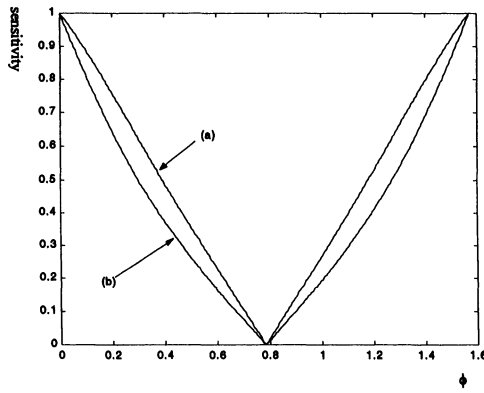


Figure 14. Sensitivity of the compatibility measures regarded as a function of ϕ : (a) volume oriented, (b) edge-oriented, ϕ is constrained to $[0, \pi/2]$.

Expressing Inclusion of Information Granules

The inclusion relation expressing an extent to which hyperbox $[\mathbf{a}]$ is included in hyperbox $[\mathbf{b}]$ is defined as a ratio of two volumes

$$\text{incl}([\mathbf{a}], [\mathbf{b}]) = \frac{V([\mathbf{a}] \cap [\mathbf{b}])}{V([\mathbf{a}])} \quad (7)$$

It is clear from the above that the inclusion measure is monotonic, non-commutative and satisfies the following boundary conditions: $\text{Incl}([\mathbf{a}], \mathbf{I})=1$ and $\text{Incl}([\mathbf{a}], \emptyset)=0$ where \mathbf{I} and \emptyset are the unit hyperbox and the empty set in \mathbf{R}^n , respectively. The calculations are straightforward; Figure 15 enumerates all cases for one-dimensional granules along with the pertinent values of this measure.

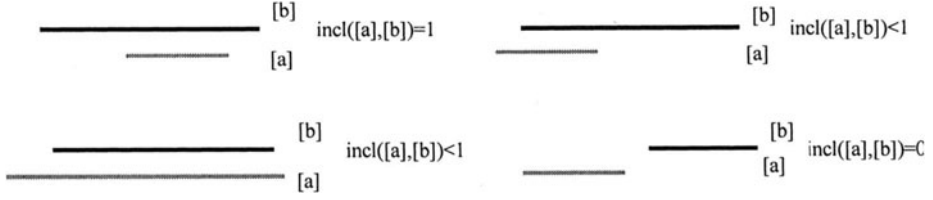


Figure 15. Computing the inclusion for interval granules [a] and [b] in R.

It is worth mentioning that the value of the inclusion measure drop down quite substantially (at a rate of r^n where $r \in \{0, 1\}$) with the increasing dimension of the space in which the information granules are distributed. For example if there is an $\frac{1}{2}$ overlap ($r=2$) in each variable in an n -dimensional space, the inclusion level expresses as 2^{-n} .

Clearly, the objective of effective information abstraction through clustering of information granules translates into identifying for which number of clusters there is a minimum overlap between the clusters. To encourage merging of clusters that have significant overlap we calculate an average of the maximum inclusion rates of each cluster in every other cluster

$$\text{overlap}(c) = \frac{1}{c-1} \sum_{i=1}^c \max_{\substack{j=1, \dots, c \\ j \neq i}} (\text{incl}([a_i], [a_j])) \quad (8)$$

where c is the current number of clusters and $[a_i]$ and $[a_j]$ are i -th and j -th cluster (granule) respectively.

It has to be pointed out however that, while the measure (7) is monotonic for any two pairs of clusters i.e. if $[a] \subset [b]$ and $[c] \subset [d]$, then $\text{incl}([a],[c]) \leq \text{incl}([b],[d])$, the change of the number and the size of clusters during the clustering process results in the collective measure, (8), having various local optima. We illustrate this effect in Figure 16.

Because of the local minima of the $\text{overlap}(\cdot)$ function it is important to have a good initial estimate of the number of clusters as a starting point for the local minimization of the function. Such an estimate is provided by our earlier analysis of the compatibility measure as discussed in the previous section.

Having accomplished the clustering process the quality of data abstraction afforded by the given set of data clusters is measured using an independent validation data set. The generality of each of the identified clusters is well quantified by the sum of the inclusion rates of the validation data items in the respective cluster.

$$\text{INCL}(i) = \sum_{j=1}^M \text{incl}([v_j], [a_i]) \quad i = 1, \dots, c \quad (9)$$

where c is the number of clusters and M is the cardinality of the validation data set.

As well as indicating whether a given cluster is representative for a large proportion of data the $\text{INCL}(\cdot)$ measure can be used to assess how representative are the training and the validation data sets. If the sets are representative, then $\text{INCL}(\cdot)$ should correlate closely with the cardinality of the individual clusters.

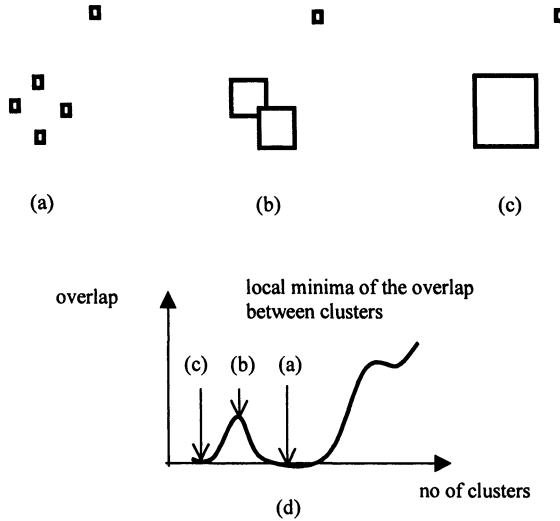


Figure 16. Progression from 5 to 2 clusters involves stage (b) during which clusters overlap. This is reflected in $\text{overlap}(3) > 0$ while $\text{overlap}(5) = 0$ and $\text{overlap}(2) = 0$.

6.5 THE GRANULAR ANALYSIS

The hyperboxes constructed during the design phase are helpful in a thorough analysis. They shed light on the nature of data as they are perceived from the standpoint of information granularity established during the design of the hyperboxes. Two main aspects are distinguished. First, we characterize the hyperboxes themselves. Second, we analyze the properties of the variables (features) forming the data space. We should emphasize that the granular analysis follows the synthesis phase and does not impact it in any way.

Characterization of Hyperboxes

The most evident characterization of the hyperboxes can be provided through their volumes, $V([b_k])$. The computations are obvious. First, we determine a ratio (normalized length)

$$\text{norm_length}_i([b_k]) = \frac{b_k^+ - b_k^-}{\text{range}_i([b_k])} \quad (10)$$

where $\text{range}_i([b_k])$ is a range of the i -th feature (variable). Since the data is normalised to a unit hypercube the $\text{range}_i([b_k]) = 1$ for all i . Second, the volume is taken as a product

$$V([b_k]) = \prod_{i=1}^n \text{norm_length}_i([b_k]) \quad (11)$$

The volume quantifies the essence of granularity of the hyperboxes. Intuitively, it states how “large” (detailed) the hyperboxes are and how much details each of them captures. One can take an average of the volumes of the hyperboxes that gives a general summary of the hyperboxes

$$\bar{v} = \frac{1}{c} \sum_{k=1}^c V([b_k]) \quad (12)$$

If one sides of the hyperbox is zero then the volume measure returns a zero value. This occurs because of the multiplicative nature of volume. To alleviate sch problem, we may also introduce a measure of an additive character. A plausible descriptor of a hyperbox could reflect a “circumference” of the hyperbox and read as follows

$$\sum_{i=1}^n \text{norm_length}_i([b_k]) \quad (13)$$

Granular Feature Analysis

The granulation of the data space (and each feature) provides an interesting insight into the nature of the variables occurring in the problem. In what follows, we provide their description in terms of sparsity and discrimination abilities. These two descriptors are exclusively implied by the granular nature of the hyperboxes.

Sparsity

When looking at a certain variable of the hyperboxes, we can visualize how much of the entire range of the variable is occupied by the hyperboxes (i.e., how *sparse* the boxes are in the given space). Take the i -th feature and calculate the sum of length of the corresponding sides of the hyperboxes that is

$$\text{tot_length}_i = \sum_{k=1}^c \text{length}_i([b_k]) \quad (14)$$

where $\text{length}_i(B(k)) = u_i(k) - l_i(k)$. The sparsity defined in the form

$$\text{sparsity}_i = \frac{\text{tot_length}_i}{\text{range}_i} \frac{1}{c} \quad (15)$$

assumes values in the unit interval. If sparsity_i is less than 1 then this represents a situation when hyperboxes (more precisely its i -th coordinate) occupy a portion of the entire range of the feature. We may state that the variable is “underutilized”. In other words, we witness a highly localized usage of this feature. The sparsity around 1 means a complete utilization of the variable. The effect of overutilization happens when sparsity achieves values higher than 1 (in this case we have some hyperboxes overlapping along this variable).

The sparsity does not capture the entire picture. A situation illustrated in Figure 17 shows two cases where the distribution of the hyperbox along the given feature is very different yet we end up having the same value of the sparsity. This leads us to another index (descriptor) that describes an overlap between the hyperboxes

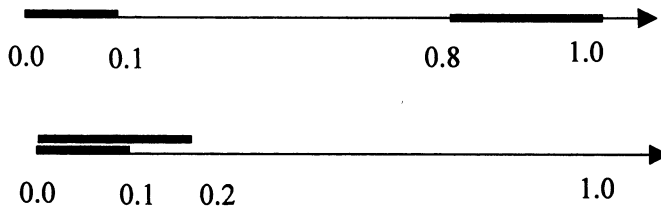


Figure 17. Two different distributions of hyperboxes (i -th feature) producing the same value of the sparsity index; in both cases the sparsity is equal to 0.3.

Overlap index

We define the following index called coordinate overlap

$$c\text{-overlap}_i = \frac{2}{c(c-1)} \sum_{k=1}^{c-1} \sum_{l>k}^c \frac{\text{length}_i([b_k] \cap [b_l])}{\text{length}_i([b_k] \cup [b_l])} \quad (16)$$

$i=1,2, \dots, n$. In this definition, $[b_k]$ and $[b_l]$ are intervals (sides) of the hyperboxes for the i -th variable. The higher the value of this index, the more overlap between the hyperboxes expressed along the given variable. When $[b_k]$ and $[b_l]$ are pairwise disjoint then the overlap is equal to zero. This means that the feature is highly discriminative as it separated the hyperboxes. The higher the overlap measure, the lower the discriminatory aspects of the feature.

Each of the measures leads to a linear ordering of the features. We can easily state which of the features is highly “utilized” and which of them comes with the most significant discriminatory properties. To form a comprehensive picture, one can localize each feature in the sparsity – overlap space. More specifically, we prefer features that exhibit low overlap (as those come with strong discriminative properties) along with low values of sparsity that points at the issue of the localized usage of the variable. It should be stressed that the above descriptors (sparsity and overlap) of the features emerge as important quantifiers because of the existence of information granules forming the hyperboxes.

6. 6 EXPERIMENTAL STUDIES

The experiments are aimed at visualizing the most essential features of granular clustering. We consider both synthetic data set and the one available on the WWW (Boston housing data).

Synthetic Data

The synthetic data sets consist of 3 groups of information granules (hyperboxes), $A_i \in [0,1] \times [0,1]$, generated by a random number generator with a uniform distribution. Each group comprises 20 granules dispersed around pre-defined points: $c_1=(0.4, 0.4)$; $c_2=(0.5, 0.6)$; and $c_3=(0.8, 0.3)$. The dispersion factor σ is varied between 0.08 and 0.15 to establish the sensitivity of the clustering process to the dispersion of the data. The clustering process is governed by the compatibility measure, (4), with the distance defined according to L_2 norm and the “compactness” factor $\alpha=0.5$.

An example of the evolution of the compatibility measure throughout the clustering process is shown in Figure 18. The intersection of the two asymptotes to the compatibility measure traced at the beginning and at the end of the clustering process indicates that 3 clusters (iteration 57) mark a natural ‘change over’ point in the behavior of the system. So, the clustering process should terminate with 3

clusters providing that the degree of overlap of clusters is also minimized for this number of clusters.

The degree of overlap of clusters was evaluated at each of the 59 iterative steps of the clustering process, according to (6), and is illustrated in Figure 19. As expected, the results of the cluster overlap analysis clearly confirm that the test data naturally falls into 3 clusters.

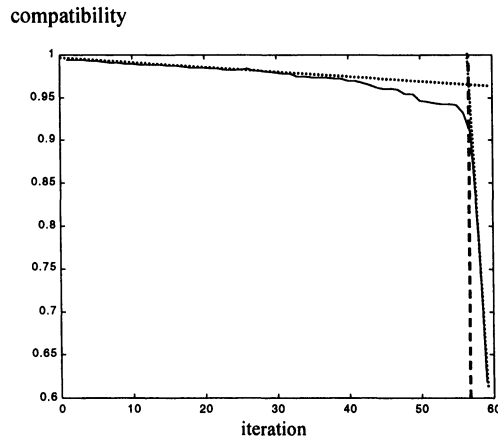


Figure 18. Compatibility measure for a single clustering process.

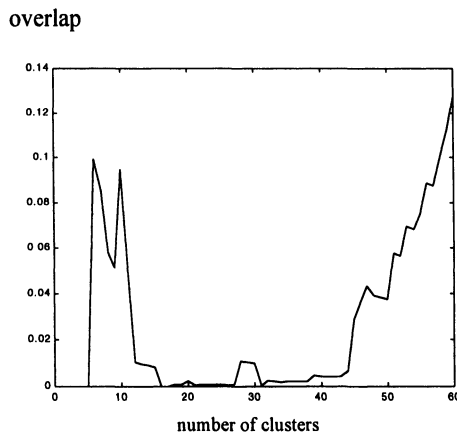


Figure 19. An average degree of overlap of clusters.

The quality of data abstraction achieved through clustering is assessed by evaluating the inclusion rate, (9), of the independently generated data set (with the same statistical properties) in the constructed clusters. An example of the output of the validation process for 10, 3 and 1 cluster is illustrated in Figure 20.

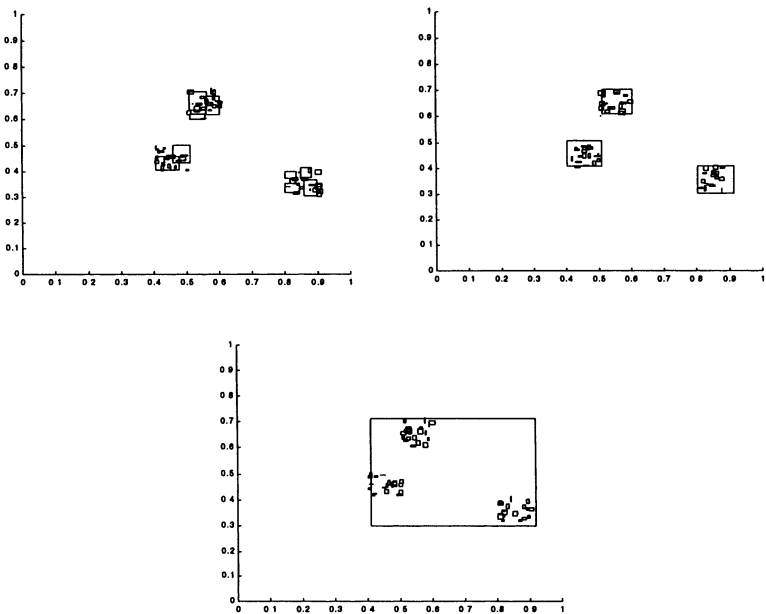


Figure 20. Inclusion of the validation data in 10, 3 and 1 cluster respectively.

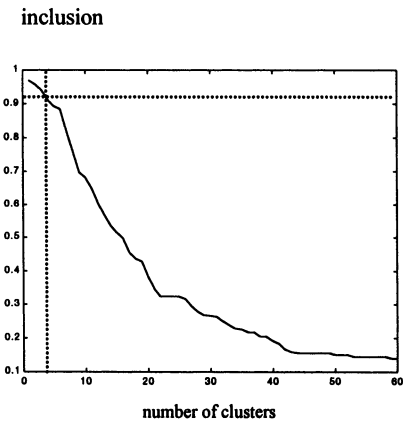


Figure 21. Average inclusion rate for the validation data set.

The change of the overall inclusion rate of the validation data throughout the clustering process is illustrated in Figure 21. It is not surprising to see that the high value of the average inclusion rate for 3 or fewer clusters confirms that 3 clusters capture the essential features of the data while the high value of the compatibility measure confirms that the clusters retain high specificity. Should the number of clusters be reduced to 2 or 1, the inclusion rate of the validation data set would only be improved marginally while there would be a very significant reduction of specificity of the cluster(s).

In order to achieve a degree of independence from the statistical characteristics of the random number generator the evaluation of the inclusion of the validation data sets in the clusters was repeated 100 times for each value of $\sigma \in \{0.08, 0.09, 0.10, 0.11, 0.12, 0.13, 0.14, 0.15\}$ and the number of clusters varying from 1 to 10. A total of 8000 training sets and 8000 validation sets were processed. Figure 22 illustrates how the inclusion measure, (5), depends on the dispersion parameter σ and the number of clusters. It is interesting to note that σ has little influence on the value of the inclusion measure. This is a very desirable characteristic of the clustering process since it suggests that the precise statistical properties of data sets do not need to be known for the clustering to be effective.

It is easy to note, from Figures 22 and 23, that the inclusion rate of 0.9 or higher is attained consistently with 3 or fewer clusters.

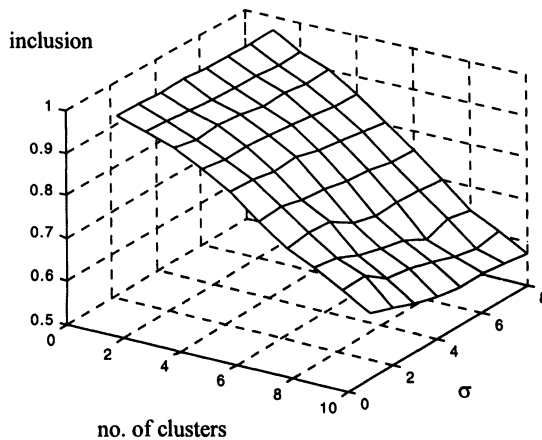


Figure 22. Average inclusion measure evaluated for 8000 training and validation sets.

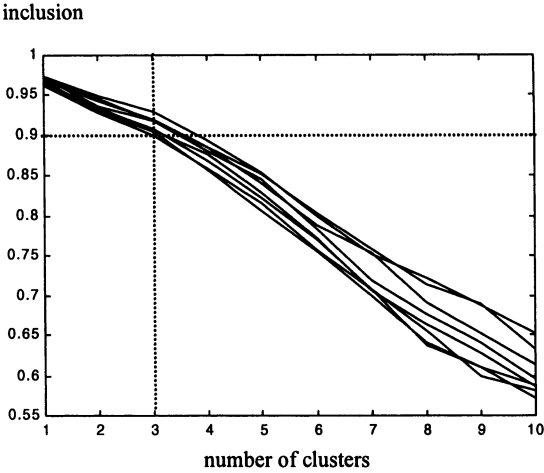


Figure 23. 2-D projection of the surface from Figure 15 resulting in a family of curves illustrating average inclusion rates of the validation data in clusters for the various values of σ .

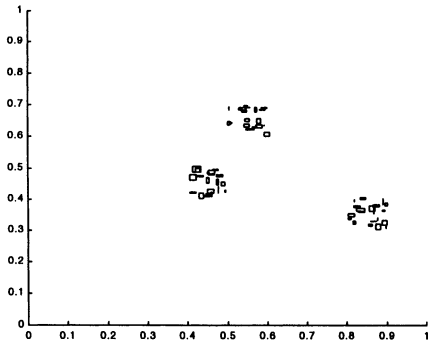


Figure 24. Two-dimensional synthetic data forming 3 distinct clusters.

Using synthetic data (3 clusters) as in Figure 24 we now assess the progression of the clustering process on the basis of the monitoring of the edge-based compatibility measure defined by (5). Figure 25 illustrates a typical evolution of the compatibility measure. As expected the asymptotical change of character of this function occurs at iteration 57 indicating that there are 3 significant clusters.

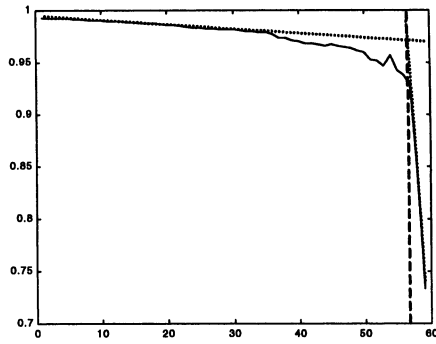


Figure 25. Edge-driven compatibility measure.

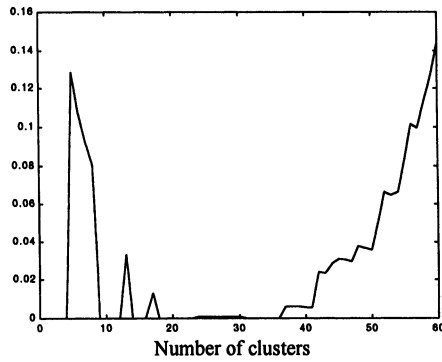


Figure 26 Average degree of overlap of clusters.

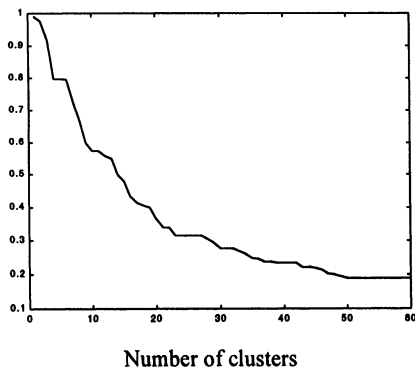


Figure 27. Average inclusion rate of the validation data in the clusters.

The results illustrated above, Figs. 26-27 are directly comparable to those contained in the earlier experiments. This result is expected bearing in mind the shape of the clusters. We examine therefore the effect of the changed topological characteristics of the clusters. A modified distribution of the patterns is illustrated in Figure 28.

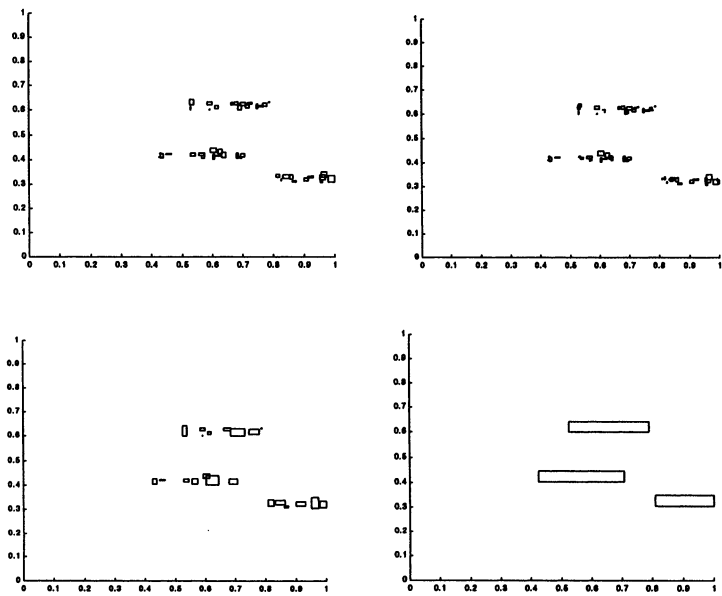


Figure 28. An alternative distribution of patterns pointing to a different ‘shape’ of clusters. Clusters are illustrated here after 1, 20, 40 and 57 clustering iterations.

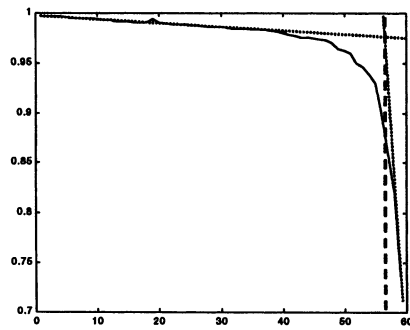


Figure 29. Compatibility measure (4) evaluated at all clustering iterations.

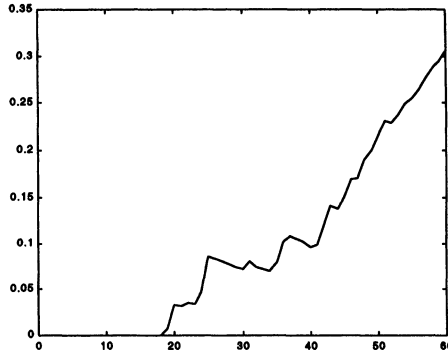


Figure 30. An average degree of overlap between clusters.

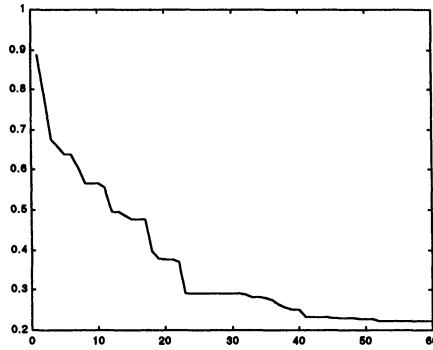


Figure 31. Average inclusion rate for the patterns in validation data set.

These results are very similar to the original clustering results that used volume-based compatibility measure (as opposed to the edge driven one).

Boston Housing Data

Although for two-dimensional data sets $B \in \mathcal{P}(\mathbb{R}^2)$ the number of clusters can be easily established by visual inspection, the higher dimensional data presents a significant challenge. We have applied therefore the algorithm to a realistic 14-dimensional data set representing factors affecting house prices in Boston area (USA). The data set has been originally compiled by Harrison and Rubinfeld, (1978), and is available from the Machine Learning Database at University of California at Irvine (<http://www.ics.uci.edu/~mllearn>). The data set comprises of 506 records.

The 14 attributes of each data record are as follows:

1. CRIM per capita crime rate by town
2. ZN proportion of residential land zoned for lots over 25,000 sq.ft.
3. INDUS proportion of non-retail business acres per town
4. CHAS Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
5. NOX nitric oxides concentration (parts per 10 million)
6. RM average number of rooms per dwelling
7. AGE proportion of owner-occupied units built prior to 1940
8. DIS weighted distances to five Boston employment centres
9. RAD index of accessibility to radial highways
10. TAX full-value property-tax rate per \$10,000
11. PTRATIO pupil-teacher ratio by town
12. B $1000(B_k - 0.63)^2$ where B_k is the proportion of blacks by town
13. LSTAT % lower status of the population
14. MEDV Median value of owner-occupied homes in \$1000's

Study A

We divided the original set into two sets. The training set, comprising 253 odd-numbered records and the validation set comprising 253 even-numbered records. It should be noted that, as a pre-processing step, all data has been mapped into a 14-dimensional unit hyperbox. The compatibility measure provided direction for the clustering process and the evolution of this measure throughout the whole process is presented in Figure 32. The gradients of the compatibility measure at the beginning and the end of the process indicate that 7 clusters represent a good abstraction of the training data.

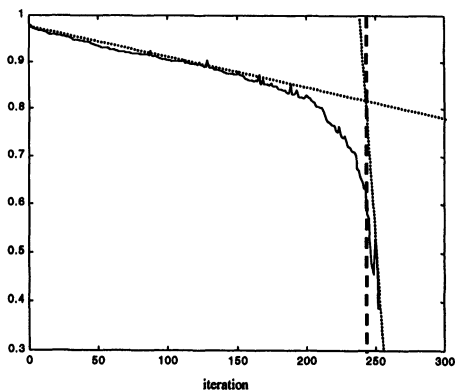


Figure 32. Compatibility measure of clusters formed from the odd-numbered records in the Boston housing data set. Iteration no. 245 corresponds to 7 clusters.

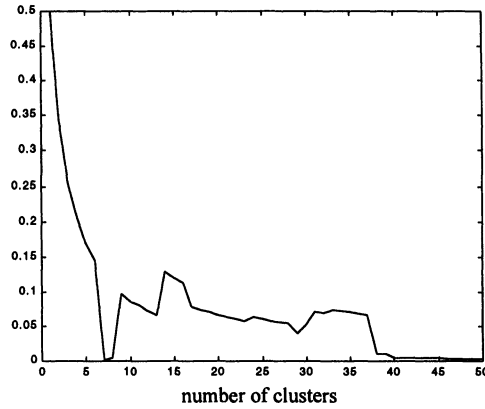


Figure 33. Degree of average overlap of clusters in the last 50 out of 252 iterations.

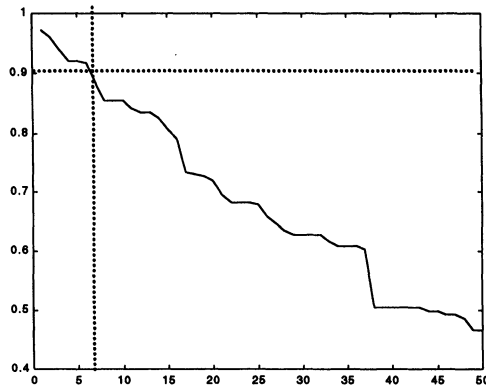


Figure 34. Average inclusion measure evaluated for 1 to 50 clusters.

In the vicinity of 7 clusters the cluster overlap indicator is minimized for 7 and 8 clusters, as shown in Figure 33. Of these two possible numbers of clusters we select the smaller number so as to achieve greater granulation of the original data. The generality of the identified clusters was tested by evaluating average inclusion of the validation data set (even-numbered records from the original data set) in the sets of clusters identified in the last 50 steps of the clustering process. This is illustrated in Figure 34. The value of over 90%, achieved for 7 clusters, indicates a good abstraction of the detailed data that is achieved with this number of clusters.

To gain a more detailed insight into the makeup of the 7 clusters we evaluated an aggregate inclusion measure (9), using the validation set, and compared the results

with the cardinality of each cluster. It is clear, from Figure 35, that out of 7 clusters 3 have a significant support in the two data sets while the other 4 clusters represent data that could be described as significant exceptions. It is interesting to note however that the zero inclusion rates of the validation data in clusters 3, 4 and 7 indicate that the small data sample makes it difficult to do a proper evaluation of the clusters. The full description of the identified clusters is given in Table 1.

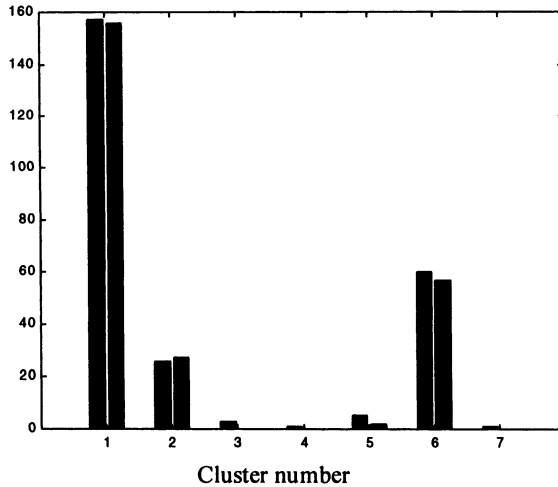


Figure 35. Cardinality (first bar) and the aggregate inclusion rate (second bar) for each of the 7 clusters.

Table 1. Description of the 7 clusters. (L_i represents minimum coordinates of the i -th hyperbox and U_i represents maximum coordinates).

Variables 1 through 7

L_1	0.0063	0	0.7399	0	0.3850	4.9730	6.0004
U_1	2.6354	95.0000	19.5800	1.0000	0.6470	8.3980	100.0000
L_2	0.0686	0	8.1399	0	0.5200	4.9030	69.6999
U_2	2.7795	0	27.7400	0	0.8710	6.4580	100.0000
L_3	1.1265	0	19.5800	1.0000	0.8710	5.0120	88.0004
U_3	3.3213	0	19.5800	1.0000	0.8710	6.1290	100.0000
L_4	2.0099	0	19.5800	0	0.6050	7.9290	96.2005
U_4	2.0099	0	19.5800	0	0.6050	7.9290	96.2005
L_5	3.4744	0	18.1001	1.0000	0.6310	5.8750	82.8997
U_5	8.9834	0	18.1001	1.0000	0.7700	8.7800	97.4997
L_6	2.3783	0	18.1001	0	0.5320	4.1380	41.9002
U_6	73.5337	0	18.1001	0	0.7700	7.0610	100.0000
L_7	88.9762	0	18.1001	0	0.6710	6.9680	91.8999
U_7	88.9762	0	18.1001	0	0.6710	6.9680	91.8999

Variables 8 through 14

L ₁	1.7984	1.0000	192.9998	12.6000	288.9906	1.9199	12.7000
U ₁	10.7103	8.0000	469.0011	22.0000	396.9000	30.8101	50.0000
L ₂	1.3459	2.0000	188.0008	14.7000	70.8002	6.4300	8.1000
U ₂	3.9900	4.9999	711.0000	21.2000	396.9000	29.6801	24.3000
L ₃	1.3216	4.9999	402.9980	14.7000	321.0184	12.1200	13.4002
U ₃	1.7494	4.9999	402.9980	14.7000	396.9000	26.8200	17.0002
L ₄	2.0459	4.9999	402.9980	14.7000	369.2980	3.7000	50.0000
U ₄	2.0459	4.9999	402.9980	14.7000	369.2980	3.7000	50.0000
L ₅	1.1296	24.0000	665.9989	20.2000	347.8787	2.9600	17.7998
U ₅	2.7227	24.0000	665.9989	20.2000	395.4287	17.5999	50.0000
L ₆	1.1370	24.0000	665.9989	20.2000	0.3200	3.2601	5.0000
U ₆	3.7240	24.0000	665.9989	20.2000	396.9000	37.9700	50.0000
L ₇	1.4165	24.0000	665.9989	20.2000	396.9000	17.2099	10.4000
U ₇	1.4165	24.0000	665.9989	20.2000	396.9000	17.2099	10.4000

The results of feature analysis is summarized in terms of their sparsity and overlap values. This analysis provides with an interesting observation about the discriminatory properties of the variables in the problem. The most dominant ones are: crime rate (1), nitric oxide concentration (5), index of accessibility to radial highways (9), and proportion of non-retail business acres (3). In other words, these are the variables that discriminate between hyperboxes (we stress that that the discriminatory aspects have been raised in the setting of the information granules rather than classes that have never been established in the first place).

Variable no.	sparsity	c-overlap
1	0.135	0.1826
2	0.136	0.7143
3	0.201	0.2194
4	0.143	0.3333
5	0.291	0.1933
6	0.326	0.3255
7	0.307	0.3759
8	0.210	0.3397
9	0.062	0.2109
10	0.218	0.2381
11	0.241	0.2234
12	0.344	0.4357
13	0.458	0.4399
14	0.426	0.3759

Study B

In order to ascertain whether the selection of records for the training and the validation data sets had influenced significantly conclusions regarding the number of clusters abstracting the original data set, we repeated the clustering process with the training and validation sets switched round. Again the compatibility measure directed the clustering process and the asymptotic evolution of the measure, at the initial and final stages of the process, indicated that 6 data clusters mark a ‘change-over’ point in the clustering process (Figure 36).

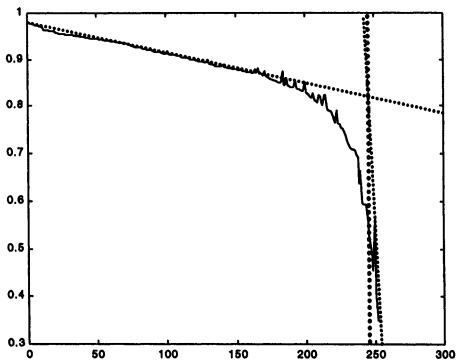


Figure 36. Compatibility measure of clusters formed from the even-numbered records in the Boston housing data set. Iteration no. 246 corresponds to 6 clusters.

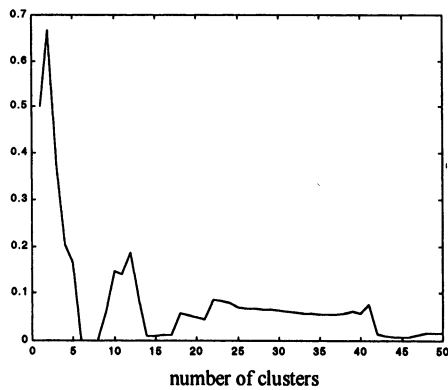


Figure 37. Degree of average overlap of clusters in the last 50 iterations.

The curve showing the average degree of overlap between the clusters, illustrated in Figure 37, indicates that a minimum overlap is achieved with 6, 7 and 8 clusters. For the ease of comparison with the Study A case we select 7 clusters for the validation

stage. The average inclusion rate of the validation data set (odd-numbered records from the original data set) in the 7 clusters is approx. 30% worse than in the previous case, averaging at 86%. This is illustrated in Figure 38.

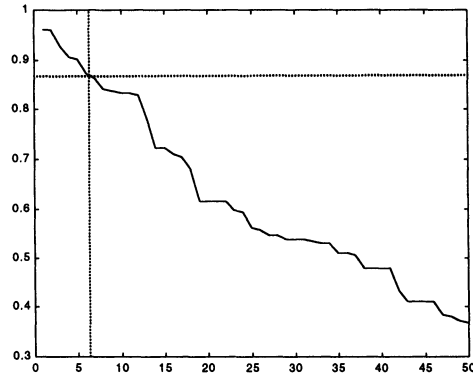


Figure 38. Inclusion measure evaluated for 1 to 50 clusters.

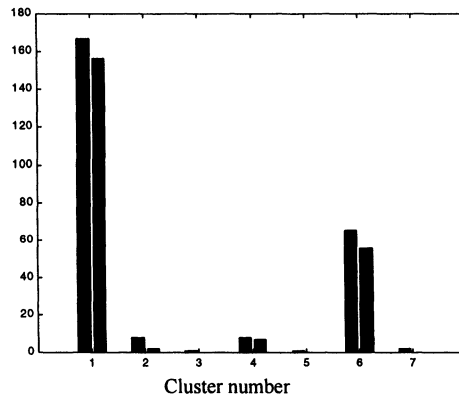


Figure 39. Cardinality (first bar) and the aggregate inclusion rate (second bar) for each of the 7 clusters

The reduction of the average inclusion rate in this case suggests that the training and validation sets contain a small number of unique patterns that do not have counterparts in the other set. The result is that although the distinctiveness of these patterns warrants their inclusion in separate clusters, the cross-comparison of these 'minority clusters' is very limited. This is further verified by the inspection of Figure 39, which shows that the clusters 3, 5 and 7 are representing by 1, 1 and 2 patterns respectively with no corresponding patterns in the validation set. It is also interesting

to note that, compared to the Study A, there is a greater discrepancy between the cardinality of the clusters and the inclusion rate. We conclude therefore that the size of the data supports only firm conclusions about 2 clusters and the characterization of further clusters requires an order of magnitude greater data sample.

The sparsity and c-overlap of the features (variables) are very similar as in Study A meaning that some global properties discovered in the data set have been retained.

Variable no.	sparsity	c-overlap
1	0.117	0.1414
2	0.229	0.2667
3	0.284	0.1432
4	0.143	0.5238
5	0.258	0.0985
6	0.348	0.3391
7	0.393	0.3144
8	0.221	0.1674
9	0.075	0.1769
10	0.155	0.1560
11	0.228	0.0760
12	0.303	0.3276
13	0.443	0.3762
14	0.412	0.3175

6. 7 CONCLUSIONS

The granular clustering approach, discussed in this chapter, provides a constructive way of forming information granules that capture the essence of large collections of numeric data. In this sense, the original data are compressed down to a few information granules whose location in the data space and granularity reflect the structure characterizing the data. The approach emphasizes the transparency of the results (hyperboxes). The way in which information granules are formed is guided by two essential properties: the distance between information granules and the size of the potential granule that arises through merging of two other granules. These two aspects are encapsulated in the form of the compatibility measure. Moreover we discussed a number of indexes describing the hyperboxes and expressing relationships between such information granules. It has been shown how to validate the granular structure. The resulting family of the information granules is a concise descriptor of the structure of the data – we may call them a granular *signature* of the data patterns. The hyperbox approach provides a good reference point for more detailed instruments of information granulation such as fuzzy sets (Kandel, 1986; Pedrycz, Gomide, 1998).

It should be stressed that the proposed approach to data analysis is *noninvasive* meaning that we have not attempted to formulate specific assumptions about the distribution of the data but rather allow the data to “speak” freely. This is accomplished in two main ways

- first, the hyperboxes are easily understood by a user as each dimension (variable) comes as a part of the construct.
- second, the approach finds relationships that are direction-free meaning that we do not distinguish between input and output variables (which could be quite restrictive as we may not know in advance what implies what). Obviously, this feature is quite common to all clustering methods

Furthermore the granulation mechanism puts the variables (features) existing in the problem in a new perspective. The two indexes such as sparsity and overlap are useful in understanding the relevance of the variables, in particular their discriminatory abilities.

REFERENCES

- Anderberg, M.R. (1973), *Cluster Analysis for Applications*, Academic Press, N. York.
- Bargiela, A. (2001), Interval and ellipsoidal uncertainty models, In: W. Pedrycz (ed.) *Granular Computing*, Physica-Verlag, 23-57.
- Bargiela, A., Arsene, C., Tanaka, M. (2002), Knowledge-based neurocomputing for operational decision support, *Knowledge-based Engineering Systems KES 2002*, Crema, Sept. 2002.
- Bargiela, A., Pedrycz, W. (2002), From numbers to information granules: A study in unsupervised learning and feature analysis, in: *Hybrid Methods in Pattern Recognition* (Bunke, H., Kandel, A. eds.), World Scientific, 75-112.
- Bargiela, A., Pedrycz, W. (2002), Recursive information granulation: Aggregation and interpretation issues, *IEEE Trans. on Syst. Man and Cybernetics*, to appear.
- Bezdek, J.C. (1981), *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, N. York, 1981.
- Gabrys, B., Bargiela, A. (2000), General fuzzy Min-Max neural network for clustering and classification, *IEEE Trans. on Neural Networks*, Vol. 11, No. 3, 769-783.
- Harrison, D., Rubinfeld, D.L. (1978), Hedonic prices and the demand for clean air, *J. Environ. Economics & Management*, vol.5, 81-102.
- Kandel, A. (1986), *Fuzzy Mathematical Techniques with Applications*, Addison-Wesley, Reading, MA.
- Kohonen, T. (1982), Self-organized formation of topologically correct feature maps, *Biological Cybernetics*, 43, 59-69.

Kohonen, T. (1995), *Self-organizing Maps*, Springer Verlag, Berlin.

Pedrycz, W. (1997), *Computational Intelligence: An Introduction*, CRC Press, Boca Raton, FL.

Pedrycz, W., Smith, M.H., Bargiela, A. (2000), Granular clustering: A granular signature of data, *Proc. 19th Int. (IEEE) Conf. NAFIPS'2000*, Atlanta, July 2000, 69-73.

Pedrycz, W., Gomide, F. (1998), *An Introduction to Fuzzy Sets*, Cambridge, MIT Press, Cambridge, MA.

Pedrycz, W. (2001), Fuzzy equalization in the construction of fuzzy sets, *Fuzzy Sets and Systems*, 119(2), 329-335.

Pedrycz, W., Bargiela, A. (2002), Granular clustering: A granular signature of data, *IEEE Trans. on Systems Man and Cybernetics*, 32, 2, 212-224.

Simpson, P.K. (1992), Fuzzy Min-Max neural networks – Part1: Classification, *IEEE Trans. on Neural Networks*, 3(5), 776-86.

Simpson, P.K. (1993), Fuzzy Min-Max neural networks – Part2: Clustering, *IEEE Trans. on Neural Networks*, 4(1), 32-45.

Tanaka, M., Mori, Y., Bargiela, A. (2002), Granulating keywords into sessions for timetabling conferences, *Soft Computing and Intelligent Systems Conference, SCIS 2002*, Tsukuba, Japan, Oct. 2002.

Zadeh, L.A. (1979), Fuzzy sets and information granularity, In: M.M. Gupta, R.K. Ragade, R.R. Yager, eds., *Advances in Fuzzy Set Theory and Applications*, North Holland, Amsterdam, 3-18.

Zadeh, L.A. (1996), Fuzzy logic = Computing with words, *IEEE Trans. on Fuzzy Systems*, 4(2), 103-111.

Zadeh, L.A. (1997), Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, *Fuzzy Sets and Systems*, 90, 111-117.

RECURSIVE INFORMATION GRANULATION

This chapter elaborates on the conceptual and algorithmic framework of information granulation. We provide a detailed algorithm of information granulation that is cast as an optimization problem reconciling two conflicting design criteria namely a specificity of information granules and their experimental relevance (coverage of numeric data). The resulting information granules are formalized in the language of set theory (interval analysis) and maximize local *information density*. The uniform treatment of data points and data intervals (hyperboxes) allows for a recursive application of the algorithm. We assess the quality of information granules through the application of FCM clustering (Fuzzy C-Means) algorithm. The algorithm is applied to two-dimensional synthetic data and experimental traffic data.

7. 1 INTRODUCTION

The ubiquitous nature of information granulation stems from the continuing human endeavor to extract and organize knowledge about the external world for the purpose of decision-making, control, system description, prediction and others. Zadeh (1979; 1996; 1997; 1999) promoted a notion of information granulation in the framework of fuzzy sets. Other formal and commonly exploited environments of information granulation deal with rough sets (Pawlak, 1991), probability (Kuipers, 1984) and set theory (interval analysis) (Bargiela, 2001; Davis, 1987; Gabrys, Bargiela, 2000; Moore, 1966; 1988). In a nutshell, information granules are treated as collections of entities (say numeric readings) that are grouped together because of their similarity, functional closeness or any other criterion that captures a feature of indistinguishability. Information granules give rise to hierarchies of cognitive entities. Depending upon the level of details one is interested in, we need information granules of different size (obviously, this term requires a formal definition). It goes without saying that information granules are *conceptual* constructs not necessarily directly implied by the needs of the physical world. In this way, information granules feature a high level of flexibility. On the other hand, they have to be anchored in the world of experimental data to reflect in some way the reality of the physical world. Put it differently: the design of information granules needs to take into consideration both the perception and the experimental evidence.

Adopting this point of view, this chapter casts the problem of information granulation in a well-defined algorithmic setting. We propose a detailed algorithmic path showing how information granules can be constructed on a basis of the existing experimental evidence. We also show how the derived information granules can be combined even further via recursive application of the algorithm so as to arrive at the higher levels of data abstraction. Bearing in mind the importance of interpretability of the resulting information granules we adopt interval analysis as a formal framework for the description of the algorithm. However, the approach discussed in this chapter can be translated and applied to other frameworks of granular computing such as fuzzy sets (here the conversion of the results hinges on an idea of representing fuzzy sets through their α -cuts (Kandel, 1986; Pedrycz, 1998; Zimmermann, 1985) that is splitting the problem into a family of set-based granulation tasks).

7.2 EXAMPLE APPLICATION DOMAINS

There are a number of representative domains where information granules can emerge as a useful vehicle to represent a given problem and make problem solving more efficient (Mani, 1999; Pedrycz, 1997; Sowa, 2000). The following three areas are among the most prominent applications of information granulation:

Granulation of time series Time series are commonly encountered in numerous practical problems (Box, 1970). There have been various approaches to the description of time series and their classification. They are carried out in the time domain and frequency domain. Prior to any detailed processing, time series are compressed in order to retain the most essential information and suppress details that are deemed redundant from the standpoint of further classification and processing. The essence of granulation of time series is to "discover" dominant components of the series. We may perceive these components as playing a role of basic conceptual blocks easily understood by humans and capturing the semantics of the underlying phenomenon. For instance, information granules may be formed as segments of consecutive samples of the signal. Then each segment may be labeled according to the configuration of the samples, say rapidly increasing signal, steady signal, slowly decreasing signal, etc, cf. (Das, 1998). Alternatively, as we propose here, one may consider granulating the time series value with its gradient (and/or higher order derivatives) in individual time instances. Note that standard sampling techniques are very specific examples of granulation of time series (as we attempt to capture a segment of a signal falling under a given sampling window by a single numeric value)

Granulation of digital images Digital images are two-dimensional relations. As far as understanding and processing of images is concerned, a crux there is to identify some higher level entities rather than being buried in a minute analysis completed at the level of individual pixels. Such tangible and semantically sound entities are

information granules. They may arise at the level of basic homogeneous regions (in terms of brightness, color and texture) one can identify in an image. These entities are inherently hierarchical: at a higher level we may think of individual objects in the image (that are composed of the granules arising at the lower level with more specific and less abstract information granules). At the technical end, the simplest and least abstract information granules are formed by defining n -by- m blocks of pixels (Madisetti, 1998; Oppenheim, 1983; 1989). At the higher level, we are concerned with various clustering techniques that help us construct abstractions out of the low-end (more detailed) information granules such as the already mentioned blocks of pixels.

Granulation of spatial structures An array of current modeling pursuits occurs in the realm of distributed systems such as networks. Obvious examples of these architectures are electric networks, water networks or telecommunication networks. In spite of their evident diversity, the networks share several profound commonalties. In particular, a hierarchical type of modeling is omnipresent there. Instead of analyzing the entire network, we split it into subnetworks (modules) that are loosely connected and proceed with a detailed analysis at this level. Obviously, this task is more tangible and manageable from the computational and interpretation standpoint. Each subnetwork is an information granule that is afterwards treated as a conceptual and algorithmic entity. For instance, when looking into a flow of traffic in a complex network, we partition the network into modules (call them telecommunications granules) and study all incoming and outgoing traffic from this perspective, refer to Figure 1. The concept of hierarchy and information granulation is inherently associated with GIS (Geographic Information Systems) systems where we anticipate various levels of detail and control the process of concentrating on specific aspects by establishing proper levels of information granularity.

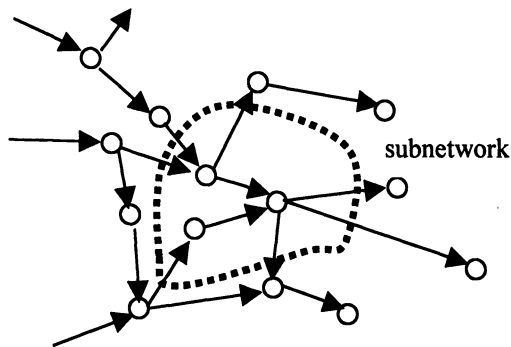


Figure 1. A concept of network granulation; we focus on structural granules by studying a flow of traffic at the level of a selected part of the network.

7. 3 INFORMATION GRANULES: DESIGN AND CHARACTERIZATION

In this section, we discuss the algorithmic layer of information granulation. As we have confined ourselves to a set-theoretic formalism, we show how to construct intervals or their multidimensional versions (that is hyperboxes). In the discussed framework, granulation applies to both numeric and granular data. The granular properties of sets are straightforward: the larger the size of the interval, the lower its granularity. The cardinality of a set, $\text{card}(\cdot)$, serves as a suitable measure of information granularity. In general, the following holds: the bigger the cardinality of the set, the lower its granularity.

Building Set-Based Information Granules

In the proposed approach, information granules are designed in two stages (phases). First, in the entire data set under analysis, we define a size of a segment (window of granulation), specify the elements (data points) within each segment and in sequel use these elements to construct a detailed form of the information granule. More formally, the granulation process can be delineated as follows

$$\mathbf{X} \xRightarrow{\Omega} A \quad (1)$$

In the above scheme, \mathbf{X} denotes an original data set, Ω is a set of disjointed time periods Ω_k representing windows of observation, and A is a set of information granules. The first phase is straightforward: by defining the size of the data segment ($\omega_0 = \text{card}(\Omega_k)$, $k=1,2,\dots,G$) we embrace a collection of data points that is of interest and needs to be considered when constructing a detailed model of the information granule. It should be emphasized that while ω_0 defines a maximum cardinality of initial granules it does not prevent formation of several smaller granules within any given window of observation (as will be explained later). Windows of observation can be formed in many different ways, refer to Figure 2. The choice depends on the problem and reflects a way in which the semantics of the problem is addressed. For instance, the window can include a fixed number of samples or it can embrace a variable number of data points that fall within a monotonic (increasing or decreasing) part of the signal. Note that the formation of the window of observation is implied by the characteristics of the problem at hand.

In the sequel, the process of constructing (and subsequent recursive refinement) of information granules comprises two phases:

1. derivation of information granule(s) from the original numeric data contained in the window of observation;
2. recursive processing of the mixture of granular and numeric data.

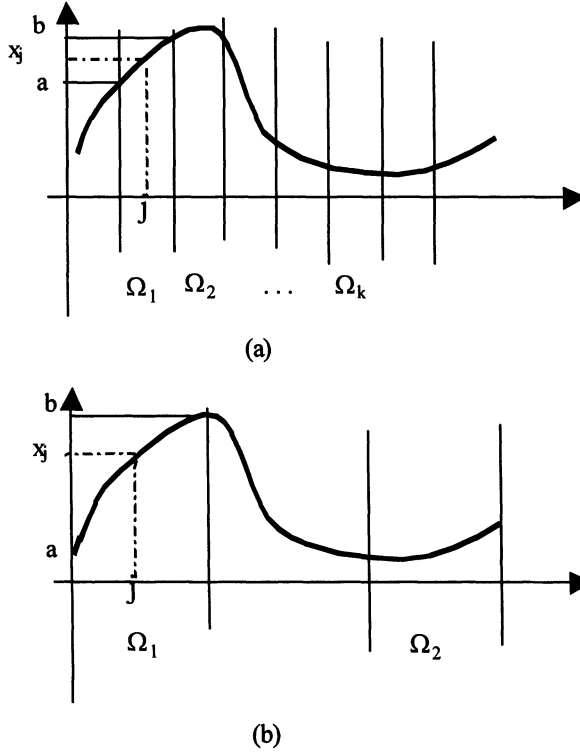


Figure 2. Examples of segments (windows of observation) Ω_k : (a) the same granularity; (b) variable granularity induced by the monotonicity of the signal.

In the detailed construct, we start with a collection of data $\{x_j: j \in \Omega_k\}$ as depicted in Figure 3(a). The phase-1 granulation results in a mixture of granules and data points that represent local maxima of the ‘information density’ function (defined later in this section). The result of this granulation is visualized in Figure 3(b). Subsequent recursive granulation of the output from phase-1 produces level-2, level-3, etc., granules (as shown in Figure 3(c)). For simplicity of discussion, we consider here just scalar numeric data but we can easily generalize this construct to a vector case by studying each coordinate of the multidimensional data separately.

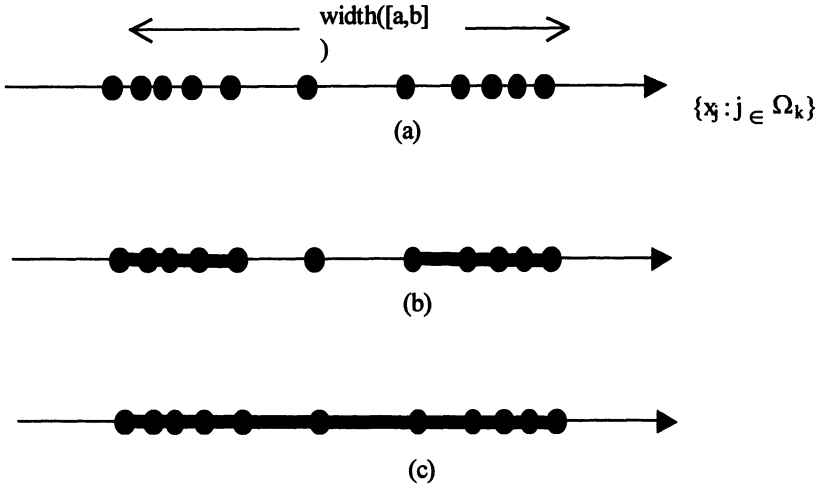


Figure 3. Illustration of the concept of recursive granulation: (a) original data, (b) phase-1 granulated data, (c) phase-2 (phase-3, etc.) granulated data.

The set theoretic framework introduced in Chapter 2 provides a convenient basis for a 'bottom-up' approach to understanding the nature of data, as advocated in (Chiu, 1996; Das, 1998; Hata, Mukaidono, 1999). In this context we discuss optimization-based granulation algorithm that does not require specification of the number or size of information granules and it focuses fully on the character of the data itself.

Building interval-valued granules arises as a compromise between two evidently conflicting requirements

- i) the interval should "embrace" as many elements of $\{x_j : j \in \Omega_k\}$ as possible (to be a sound representation of the window of observation)
- ii) the interval should be highly specific. This translates into the requirement of a minimal length of this interval (set).

As far as the first requirement is concerned, a cardinality of the set covering elements of Ω_k is a suitable criterion, that is

$$\text{card}(I) = \sum_{x_i \in X} \chi_{[a,b]}(x_k) \quad (2)$$

where $I = [a,b]$ denotes the interval we are about to construct and $\chi_{[a,b]}$ stands for its characteristic function, that is

$$\chi_{[a,b]}(x) = \begin{cases} 1, & \text{if } x \text{ is in } [a, b] \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

The specificity of the interval can be directly associated with its width,

$$\text{width}(I) = \text{width}([a, b]) = b - a \quad (4)$$

More precisely, the larger the width of the interval, the lower its specificity. While this definition is straightforward, we will be using its slightly enhanced version expressed as

$$\phi(\text{width}([a, b])) \quad (5)$$

where " ϕ " is monotonically increasing function of the original width and $\phi(0) = 1$ (as will become obvious in a while, this boundary condition is introduced for the sake of uniformity of processing of data points and data intervals). For instance, a mapping of interest can assume the form

$$\phi(u) = \exp(u) \quad (6)$$

Bearing in mind a conflicting nature of the requirements (i) - (ii) that is captured in the form

$$\text{card}(I) \rightarrow \max \quad \phi(\text{width}([a, b])) \rightarrow \min \quad (7)$$

it is legitimate to take a ratio of these expressions

$$\sigma = \frac{\text{card}(I)}{\phi(\text{width}(I))} \quad (8)$$

and determine the interval I so that it maximizes expression (8). In this way, we cope simultaneously with the two contributing optimization problems defined in (7). We refer to the optimization expressed by (8) as maximization of 'information density' of granules. This is to distinguish it from the concept of 'data density' that is typically represented as a ratio of cardinality of a given set over the volume of the pattern space containing this set. Consequently 'data density' is not defined for a single numeric data (zero volume in pattern space).

The choice of function $\phi(u)$ depends on the preference for large or small information granules. Figure 4 shows contour plots of the expression (8) obtained with $\phi(u)$ defined as in (6). It can be seen that the decrease of the gradient of the contours with the increase of the cardinality of the granules implies inherent preference for smaller

granules. This is an advantageous feature as it gives us a possibility of avoiding undue influence of inherently local optimization on the more global view of data that is obtained through recursive application of the granulation algorithm. An alternative choice of $\phi(u)=1+u$ results in constant-gradient contours of (8) and is thus less appropriate in the context of our algorithm. A function $\phi(u)=(1+u)^2$ results in contour plots that are broadly similar to those obtained with $\phi(u)=\exp(u)$ but it is less convenient numerically.

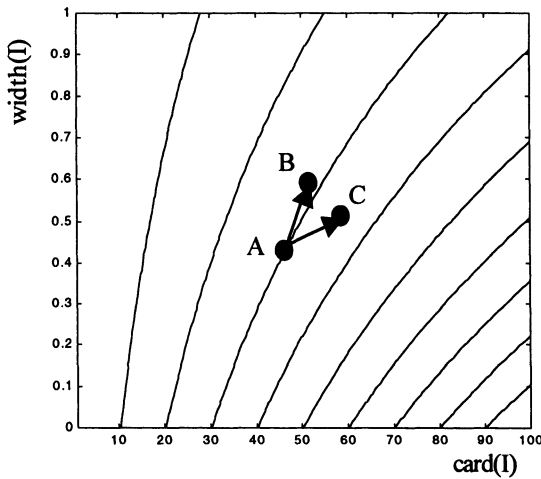


Figure 4. Contour plot of the information density function (8); $\sigma(I)=\text{const.}$. Transition from granule A to B represents a net decrease of information density and is therefore avoided. Transition from A to C represents formation of a granule with higher information density.

The above considerations generalize easily on the case of multi-dimensional data. The maximization of information density, implied by the expression (8), can be performed for multi-dimensional hyperboxes. We consider in this case a ratio of the cardinality of the input data set contained in such hyperboxes to a function of volume of the hyperboxes. However, such a direct approach creates dependence of the information density measure on the dimensionality of the pattern space. Given that in the interest of the uniformity of processing of data points and intervals we deliberately increase the dimensionality of the pattern space (as explained in detail in Section 4) it is advantageous to consider a dimensionality-invariant version of mapping $\phi(u)$. This can be given as follows

$$\phi(u) = \exp(\max_i (u_i)) * \exp(\max_i (u_i) - \min_j (u_j)) \quad (9)$$

where $u=(u_1 \ u_2 \ \dots \ u_n)$, $u_i = \text{width}([a_i, b_i])$ and $i,j=1,2,\dots,n$, is an index of the dimension of the pattern space. The first exponent function in (9) ensures that the specificity of information granules is maximized through the reduction of the maximum width of the hypercube along all dimensions in the pattern space. The second exponent in (9) ensures that the hyperboxes are as similar to hypercubes as possible. The above function can be expressed in a more compact form

$$\phi(u) = \exp(2 * \max_i (u_i) - \min_j (u_j)) \quad (10)$$

where, $i,j=1,\dots,n$. It is clear that (10) is not affected by the dimensionality of the pattern space. The maximization of the width of the hyperbox (granule), over all dimensions of the pattern space, results in a scalar value that is of the same order regardless of the space dimension. Also, the function satisfies the original boundary condition $\phi(0) = 1$, since for the point-size data $\max_i (u_i) = \min_j (u_j) = 0$, $i=1,\dots,n$.

While, in general, the pattern space \mathbf{X} can be any subset of \mathbf{R}^n , we restrict the operation of the optimization task (8) to a unit hypercube $[0, 1]^n$. Such a restriction does not imply any loss of generality of our approach while affording clear computational benefits (with regard to mapping $\phi(u)$).

The optimization-based granulation of data is carried out as a one-pass simulation process:

- a) Normalize data to a unit hypercube;
- b) Initialize data structures representing cardinality and the width of individual data items (1 and 0 respectively for the point-data);
- c) Calculate and store the value of 'information density' (as implied by (8)) of hypothetical granules formed by any two data items in the input data set. This forms an upper-diagonal matrix D of size $N \times N$, where N is the cardinality of the input data set.
- d) Find a maximum entry in D ;
- e) If the maximum corresponds to an off-diagonal element (i -th and j -th coord):
 - merge the two information items (identified by the i -th row and j -th column) into a single information granule, which has width defined by the maximum and minimum values of coordinates in each dimension from the two component granules
 - update the cardinality of the resulting granule to the sum of the cardinality counts of the component granules;
 - update the i -th row and column of D with the information pertinent to the newly formed information granule and remove the j -th row and column from D ;
 - return to d)

- f) If the maximum corresponds to a diagonal element ($i=j$):
- copy the granule to an output list and remove the corresponding row and column from matrix D;
 - if the size of matrix D is greater than 1, return to d), otherwise terminate.

Computational complexity of this granulation algorithm is $O(N^2)$ owing to the computations of matrix D in step c). However, unlike the clustering techniques (such as FCM, (Bezdek, 1981)), the granulation process has an inherently local character and can be easily applied to a partitioned input data thus circumventing the high computational cost associated with large data sets. It is worth pointing out that the size of matrix D is being reduced in every iterative step by one row and one column thus the number of steps equals $N-1$.

The above algorithm is somewhat similar to ‘subtractive clustering’ proposed by Chiu (1996) in that the algorithm avoids any arbitrary partition of the input space and is driven purely by the existing input data. In contrast to grid-based methods, areas of input space that do not have data are simply ignored by the two algorithms. Both this algorithm and ‘subtractive clustering’ avoid combinatorial explosion of relationships with the increasing dimension of the input space. Since the algorithms maintain linear computational complexity with respect of the input space dimension (not to be confused with the complexity with respect of the cardinality of the data set which is $O(N^2)$), they are particularly suitable for processing multi-dimensional data. Another common characteristics of the two algorithms is that they maintain a localized view of data. As the granulation proceeds, the identified clusters do not exercise further influence on data points that remain after their removal.

However, there is a significant difference between the two algorithms. The algorithm presented here does not make any assumptions about the maximum size of granules. Granules are allowed to grow as long as their local data density keeps increasing. Also, we do not make any arbitrary decision about the separation of cluster centers. The formation of closely separated granules is largely avoided by the very nature of maximization of information density, which tends to increase the size of granule if it means adding sufficiently large number of data items (another granule) without undue increase of its volume. If, on the other hand, the increase in volume would imply the reduction of information density, the granule does not expand and remains well separated from the neighboring granules. Another distinguishing feature of our algorithm is that it allows processing both point-size and hyperbox data. This is an important characteristic that allows hierarchical granulation of data. It should be noted that hierarchical granulation enables overcoming the limitations of the ‘local view’ of data while supporting the application of the algorithm to a partitioned input data set.

It is also instructive to point out an important difference between the hierarchical clustering and the hierarchical granulation proposed here. In hierarchical clustering the similarity or proximity measure is evaluated for all data in the unpartitioned pattern space. This renders hierarchical clustering computationally expensive at the early clustering stages. By contrast, hierarchical granulation can operate on the partitioned pattern space (thus achieving significant computational gains because of the quadratic computational complexity of the algorithm with respect to the cardinality of the data set) and the subsequent application of the algorithm to the partially granulated patterns enables arriving at the globally optimal granulation.

The granulation algorithm introduced here is developed against a background of some recent developments in this area. In particular it complements the fuzzy granulation approaches proposed in (Thiele, 2000; Berthold, Huber, 1999; Hata, Mukaidono, 1999). We demonstrate that crisp granulation followed by fuzzy clustering offers a powerful framework for deriving data abstractions. To illustrate the operation of the granulation algorithm we consider here a synthetic 2-dimensional time series as presented in Figure 5.

The granulation algorithm is then applied to data from Figure 5(b). Apart from identifying the granules themselves we monitor the value of information density index throughout the granulation process. Of course the information density of granules identified towards the end of the process is lower than the information density of the early granules. This is because the removal of ‘high information density’ granules leaves effective voids in the pattern space. We can utilize this indicator in two ways. Either, we can terminate the granulation process when the information density of granules reaches a pre-specified threshold level (which effectively discards some data) or, we can perform a ‘higher-level’ granulation on the identified granules (which merges granules from the previous level). We adopt here the latter approach.

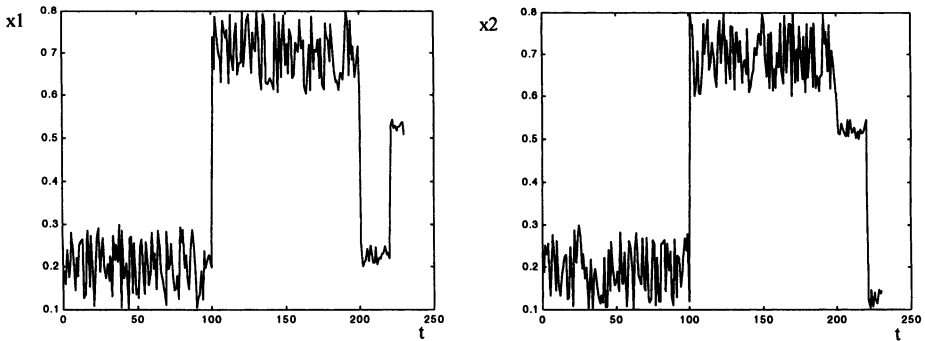
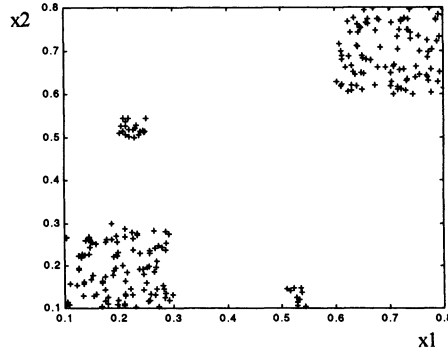


Figure 5(a). Synthetic 2-dimensional time series: time plot.



(b)

Figure 5(b). Synthetic 2-dimensional time series: state plot.

The operation of the granulation algorithm, over three hierarchical levels of granulation, is illustrated in Figure 6. The ‘level-one’ granulation compresses the original set of 230 data points into 27 granules. These granules are presented as input data to ‘level-two’ granulation, which results in 9 granules. ‘Level-three’ granulation reduces this number further to 6 granules. It is self-evident that the hierarchy of granules forms an abstraction that preserves the essential characteristics of the original data (that of four relationships in the pattern space). Even more importantly the granulation has balanced the relative count of data items in large and small data groupings thus helping smaller data groups to be ‘noticed’ in subsequent processing (clustering).

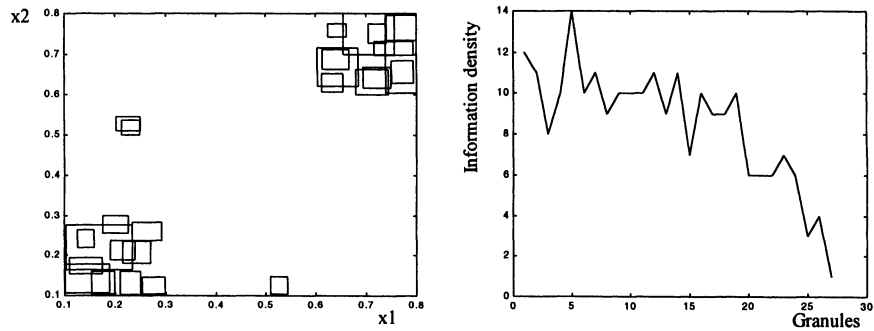


Figure 6(a). Illustration of the operation of the granulation algorithm applied to synthetic 2-dimensional time series: ‘first-level’ granulation;

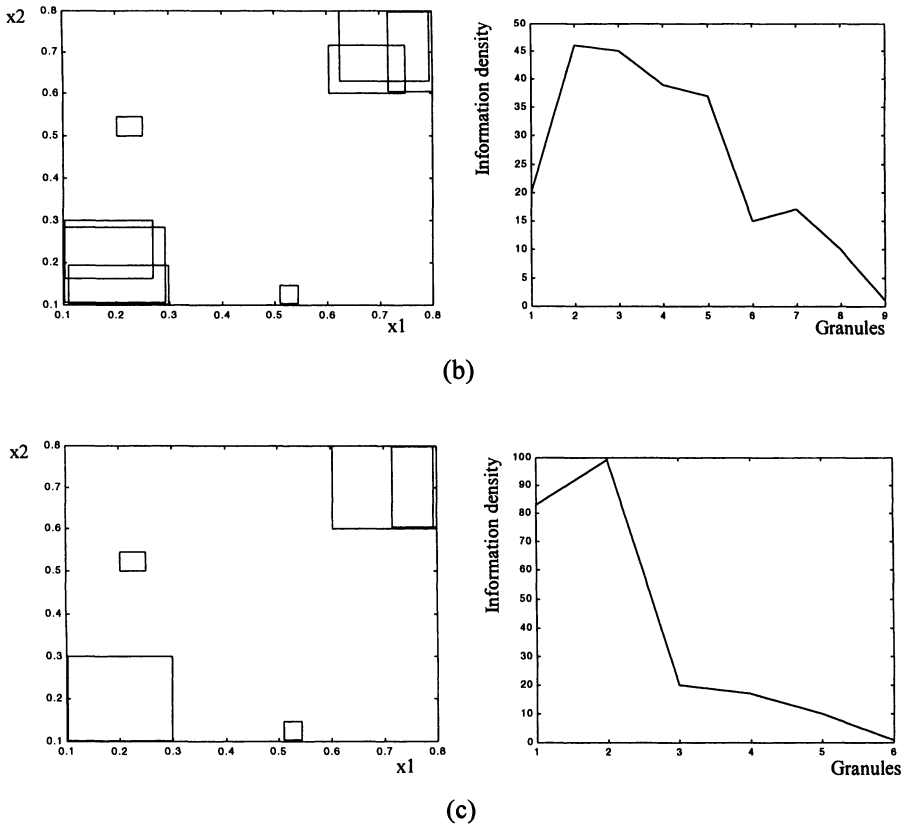


Figure 6. Illustration of the operation of the granulation algorithm applied to synthetic 2-dimensional time series:
(b) 'second-level' granulation and (c) 'third-level' granulation.

It is worth noting that the number of granulation levels does not need to be defined in advance. The hierarchical granulation is simply carried out until the number of granules identified at the subsequent granulation levels does not change. Of course, in any practical application the maximum size of granules is frequently pre-defined so that the granules map conveniently onto some linguistic entities. In this case the relative weighing of the two components in the expression (8) can be adjusted so as to achieve the required granularity.

7. 4 ASSESSMENT AND INTERPRETATION OF INFORMATION GRANULES THROUGH FUZZY CLUSTERING

The recursive application of the granulation algorithm discussed in the previous section, condense the data quite significantly. What is of fundamental interest though is whether this 'condensing' maintains the view of the essential characteristics of data. We assess here the quality of granulation by identifying a limited number of representatives of both the original numeric data and the constructed information granules. This is accomplished by clustering and identifying prototypes (representatives) of the granules, cf (Everitt, 1974). In particular, a fuzzy clustering method - a well-known FCM algorithm (Bezdek, 1981; Hoppner et al, 1999) is of interest here. As a result of this clustering mechanism, the method returns a partition matrix. This matrix captures all granules in the form of some generalized architecture of fuzzy sets formed over the family of the original information granules.

Note that in contrast to the "standard" clustering method, we are concerned with a collection of hypercubes -- sets in \mathbb{R}^n , (see Figure 7). As in any clustering pursuit, our objective is to reveal a structure in a set of these granular data. As a consequence of the granular nature of the data set, we anticipate that the prototypes returned by the FCM are also information granules. Owing to the granular nature of the data to be clustered, they need to be represented (encoded) in such a way that their aspect of granularity can be properly captured by the FCM method. A parametric method of processing heterogeneous data is a sound solution to this problem. Within this scope, several directions could be sought, refer again to Figure 7.

- (i) representing the lower and upper bound of each coordinate (feature) of the information granule (we refer to it as a bound encoding). Thus for the n -dimensional information granule we end up with a $2n$ -dimensional space of objects \tilde{x} to be clustered

$$\tilde{x} = [x_1^- \ x_1^+ \ \dots \ x_i^- \ x_i^+ \ \dots \ x_n^- \ x_n^+]$$

- (ii) representing each coordinate by a center point of the granule and its width (center-width encoding). Again, this form of representation gives rise to the $2n$ -dimensional space

$$\tilde{x} = [m_1 \ \delta_1 \ \dots \ m_i \ \delta_i \ \dots \ m_n \ \delta_n]$$

This representation is suitable if we have an interval that is distributed symmetrically around the center. Otherwise, one has to incorporate the lower and upper width. In this case, such representation implies a higher dimensionality of the space in which the clustering takes place.

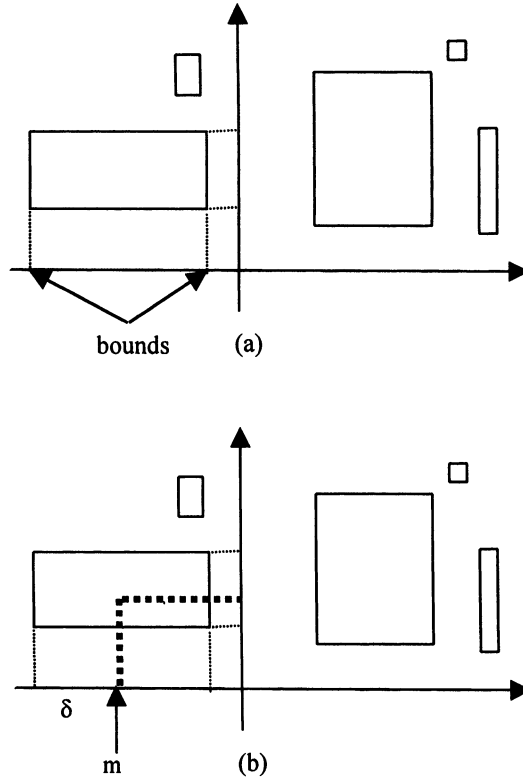


Figure 7. Information granules to be clustered and their representation
(a) bound representation and (b) center-width (m - δ) representation.

The topological implications of increasing the dimensionality of the pattern space in the above two representations can be appreciated by analyzing a single dimension of a granule. We consider here three intervals I^A , I^B and $I^C \subset \mathbf{R}^1$ and a point $P^D \in \mathbf{R}^1$ as shown at the top of Figure 8.

The bounds-encoding method generates points in \mathbf{R}^2 that have their first coordinate (x_{i-}) representing the lower bound of the interval and the second coordinate (x_{i+}) representing the upper bound of the interval. It is clear that all points in \mathbf{R}^1 , map onto points along the line $x_i^- = x_i^+$ in \mathbf{R}^2 and all intervals in \mathbf{R}^1 map onto points in a half-space $\{x_i = (x_i^-, x_i^+): x_i^+ > x_i^-, x_i \in \mathbf{R}^2\}$. What has been achieved therefore is that a heterogeneous mix of intervals and points in \mathbf{R}^1 has been converted into a homogeneous set of points in \mathbf{R}^2 ; (to be precise we are only concerned with the unit interval $[0, 1]$ and a unit box $[0, 1] \times [0, 1]$). An interesting feature of the bounds-encoding mapping is that the occurrence of inclusion/overlap of intervals is easily

detected in the mapped image in \mathbf{R}^2 . The symmetrical reflection of the mapped intervals with respect of the diagonal line, $x_i^- = x_i^+$, gives rise to a 'box' (as illustrated in Figure 8) for each interval. The boxes for disjoint intervals are disjoint and the boxes for overlapping intervals overlap as well.

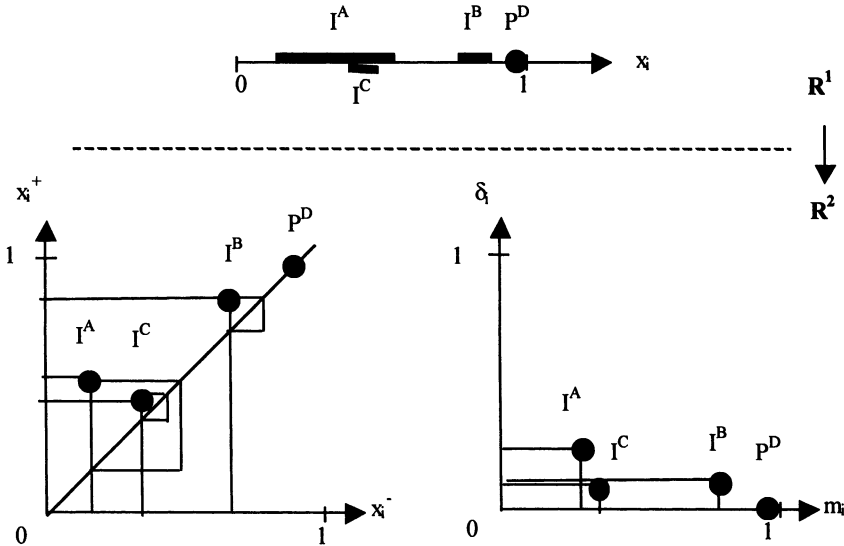


Figure 8. Mapping of three intervals and a point in \mathbf{R}^1 to \mathbf{R}^2 space using bounds- and center-width encoding.

The center-width mapping also achieves a conversion of heterogeneous mix of intervals and points in \mathbf{R}^1 , into a homogeneous set of points in \mathbf{R}^2 . However, the topological property of interval overlap is more difficult to identify in the mapped image. This is because the character of the two component dimensions is quite different. One dimension represents a value of data and the other a relative variation from this value.

Bearing this in mind, we adopt here the bounds-encoding method so that we can maintain topological interpretability of the mapped points and intervals. It should be pointed out that in general, we will concern ourselves with mapping from \mathbf{R}^n to \mathbf{R}^{2n} (or more precisely from $[0, 1]^n$ to $[0, 1]^{2n}$). Having achieved a homogeneous representation of input data, the application of standard FCM clustering, (Bezdek, 1981), returns a partition matrix and a collection of cluster prototypes. These prototypes are of the same dimensionality as the input data, thus they can be interpreted in the original data space as hyperboxes. In particular, the prototypes represent now fuzzy decomposable relations in the feature space (Pedrycz, 1998; Zadeh, 1999; Zadeh, Kacprzyk, 1999) in addition to representing, through the

partition matrix, the fuzzy membership of data in clusters. The combination of the two aspects delivers a more comprehensive insight into the granular nature of information being summarized by the prototypes.

To illustrate the clustering of information granules, we continue with the example given in the previous section. The FCM algorithm is deployed first on the original data (to provide a base reference) and then on the granulated data. The number of clusters is kept constant ($c=4$), so that the issue of size and positioning of prototypes is brought into sharper focus. The results are shown in Figure 9 and Table 1.

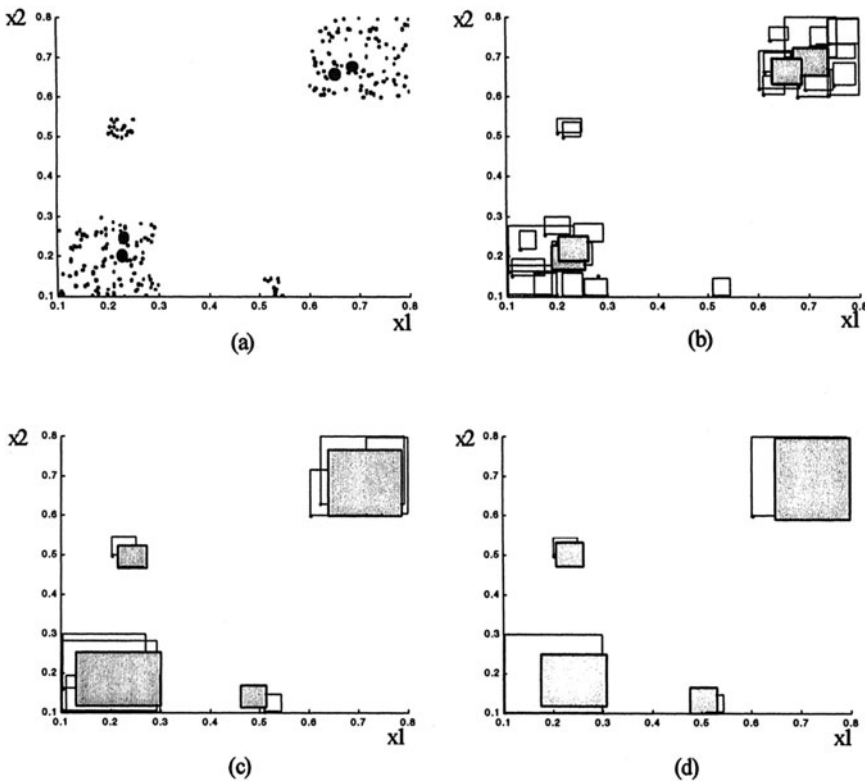


Figure 9. Clustering of granular data performed using FCM.
 (a) reference case of clustering the original 230 data points (the size of prototypes has been exaggerated for the sake of the clarity);
 (b) clustering of level-one granulated data (27 granules),
 (c) clustering of level-two granulated data (9 granules),
 (d) clustering of level-three granulated data (6 granules).

Case	Prototype	Prototype coordinates			
Reference case – original data	P1	0.6869	0.6811	0.6869	0.6811
	P2	0.6536	0.6593	0.6536	0.6593
	P3	0.2255	0.2033	0.2255	0.2033
	P4	0.2324	0.2482	0.2324	0.2482
Level-one granulated data	P1	0.1987	0.1873	0.2518	0.2381
	P2	0.6772	0.6538	0.7364	0.7226
	P3	0.2013	0.1966	0.2548	0.2470
	P4	0.6275	0.6328	0.6859	0.6982
Level-two granulated data	P1	0.2124	0.4718	0.2659	0.5214
	P2	0.4652	0.1206	0.5042	0.1656
	P3	0.6422	0.6086	0.7769	0.7674
	P4	0.1293	0.1290	0.2902	0.2501
Level-three granulated data	P1	0.1792	0.1267	0.2965	0.2446
	P2	0.4829	0.1113	0.5199	0.1572
	P3	0.6546	0.5983	0.7924	0.7935
	P4	0.2061	0.4821	0.2575	0.5312

Table 1. Coordinates of FCM prototypes as illustrated in Figure 9.

As expected, the granular input data gave rise to granular prototypes. The size of the prototypes affords a good appreciation of the spatial dimensions of the original data groupings. This is in contrast to the standard result (Figure 9(a)) where the prototypes are point-size and, in themselves, do not convey this information. Although the FCM partition matrix contains information that represents the area of influence of individual clusters its direct interpretation is quite difficult due to the complex topology of the contour plots of the partition matrix. In this sense, clustering of granular data affords a better insight into the nature of data.

Another important observation that can be made on the above results is that the information granulation helps to overcome the well-known bias of the FCM algorithm, that of under representing smaller groupings of data. Since the granulation reduces the number of information items in the high data density areas, the relative count of granules in large and smaller groupings of data evens out. In other words, granulation substitutes explicit enumeration (that unduly affects FCM) with an update of the cardinality attribute associated with individual granules (that is transparent to FCM). It can be seen that the clustering of level-two and level-three granulated data (Figure 9(c)-(d)) does not have any problems associating prototypes with the two smaller data groupings. This is a significant result that illustrates how data granulation complements fuzzy clustering.

7.5 GRANULAR TIME SERIES

The concept of information granulation opens up a new avenue of signal processing both in terms of signal representation and modeling relationships between the granular entities. In this section we look at two possible approaches to capturing the dynamics of time series and discuss it based on the granulation approach described in the previous sections.

Time-Domain Granulation

The state-space granulation (and subsequent clustering) described in the previous two sections is now compared to the direct approach in which the information granules are formed by pre-defined sets of consecutive elements of the time series. This approach is referred to as time-domain granulation. The simplest strategy within this approach is to define a ‘window of observation’ and to evaluate an appropriate granular representative within such a segment of time series. Figure 10 illustrates the principle of this approach.

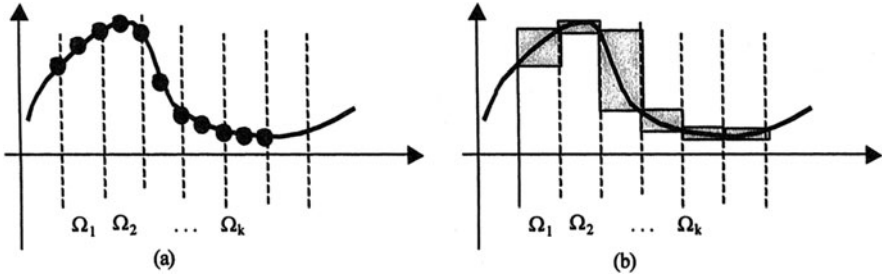


Figure 10. A simple time-domain granulation:
(a) time series; (b) information granules formed with $\omega_o=3$.

Bearing in mind our earlier comments on the bounds-encoding of information granules, we can formalize the time-domain granulation as a mapping of the original data set $\mathbf{X} = \{x^1, x^2, \dots, x^N\}$ onto a set of intervals $\mathbf{I} = \{I^1, I^2, \dots, I^G\}$ where N is a number of elements in the time series and G is the number of granules. Individual granules I^k are described as follows

$$I^k = \left(\min_{j \in \Omega_k} (x^j), \max_{j \in \Omega_k} (x^j) \right) \quad (11)$$

where

$$\Omega_k = \{i : \omega_o(k-1) + 1 \leq i \leq \omega_o k\} \quad (12)$$

and $k=1,2, \dots, G$, $\omega_0 G \leq N$, ω_0 is a granulation window. The expressions (11)(12) are easily generalized to multi-dimensional time series by applying the min- and max- operations to all coordinates of the original data. In which case we have

$$I^k = \left(\min_{j \in \Omega_k} (x_1^j), \max_{j \in \Omega_k} (x_1^j), \min_{j \in \Omega_k} (x_2^j), \max_{j \in \Omega_k} (x_2^j), \dots, \min_{j \in \Omega_k} (x_n^j), \max_{j \in \Omega_k} (x_n^j) \right) \quad (13)$$

It is clear that the mapping of \mathbf{X} onto \mathbf{I} involves the increase of dimensionality of the pattern space from \mathbf{R}^n to \mathbf{R}^{2n} , where n is a dimension of patterns. The set of intervals \mathbf{I} represents now a granulated information from the original time series. As such, the intervals I^k can be used to extract specific knowledge about the system at the higher level of abstraction compared to the one afforded with the original time series. In order to check the effectiveness of knowledge abstraction based on the information granules (13), we apply the time-domain granulation to the synthetic data of Figure 5. To make a fair comparison of our algorithm with time-domain granulation we select ω_0 to be 8, 25 and 38 so as to ensure that the granulation returns 27, 9 and 6 granules respectively. However, we start first with a granulation windows ω_0 equal 2, 3, 5 and 6, which imply formation of 115, 76, 46 and 38 granules. Results of the granulation and subsequent FCM clustering (4 clusters) are presented in Figure 11.

One thing that is immediately obvious from these results is that time-domain granulation is very sensitive to the selection of ω_0 . If the window of observation is well aligned with the boundaries of significant changes in the time series, as is the case for $\omega_0=2$ and $\omega_0=5$, the resulting granulation gives a good abstraction of the original data and the FCM clustering identifies prototypes that represent well the data. However, in a more typical case, when the window of observation includes data that belongs to two different data groupings, the time-domain granulation generates large, unrepresentative granules that adversely affect subsequent FCM clustering. Such performance is, in fact, to be expected since the transition of the time-series from one data grouping to another represents a high frequency signal that is not matched by the sampling frequency of the granulation window (as defined by the inverse of the window's width). So, the result is an irretrievable loss of information (Shannon theorem, (Oppenheim, 1983, 1989) that demonstrates itself here through large, low-specificity information granules. The FCM prototypes that are build on such granules have also low specificity and they occupy most of the pattern space (Figure 11(d)-(f)). We conclude therefore that a simple time-domain granulation should be avoided if there is no additional information available concerning the appropriateness of width of the specific granulation window.

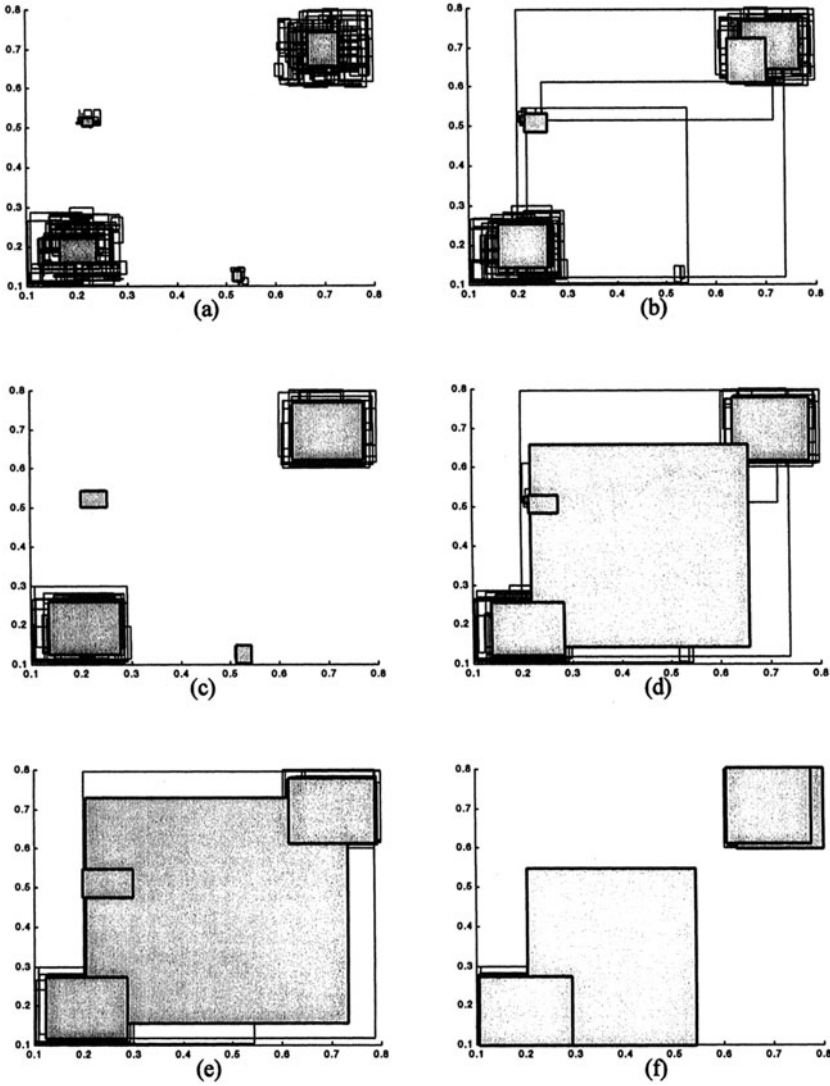


Figure 11. Results of time-domain granulation and FCM clustering
 (a) $\omega_0=2$ (115 granules); (b) $\omega_0=3$ (76 granules); (c) $\omega_0=5$ (46 granules);
 (d) $\omega_0=6$ (38 granules); (e) $\omega_0=8$ (28 granules); (f) $\omega_0=25$ (9 granules).

A refinement of the simple time series granulation approach has been proposed by Das et al. (1998). The extended method considers fixed-length subsequences of the series rather than just individual data items (Figure 12). The sub-sequences are

represented as data-points in the augmented input space that has dimension defined by the length of the sub-sequences (ω_0). Sub-sequences that have similar 'shape' are represented as nearby points in the augmented space and can be clustered using some appropriately defined distance function. However, because the property of shape similarity should be independent from the actual values of time series, the sub-sequences need to be normalized to a fixed range (typically $[0, 1]$) before they are clustered. It is worth noting that every change of the width of the granulation window (ω_0) implies the need for a re-normalization of the sub-sequences. The clustering process can be seen as a formation of a "vocabulary" (codebook) of information granules that are viewed as conceptual entities aimed at capturing the original numeric signal, see also (Pedrycz, 2001).

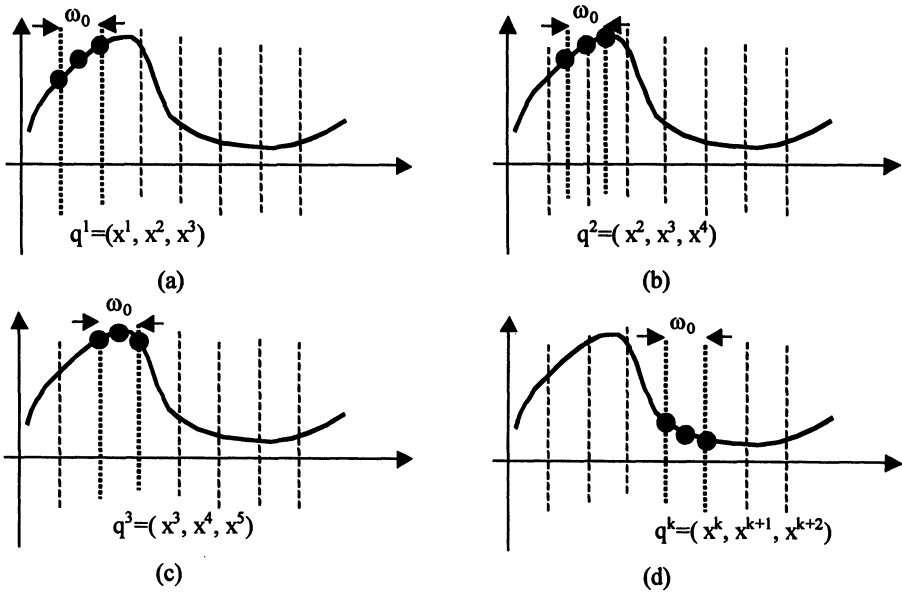


Figure 12. Granulation of time series $x^i \in \mathbb{R}^n$ ($i=1,2,\dots,N$) into fixed-length subsequences $q^j \in \mathbb{R}^{(\omega_1)^n}$ ($j=1,2,\dots,N-\omega_0+1$); $\omega_0=3$; $\omega_1=4$.

Clearly, the approach generalizes to multi-dimensional time series. In this case the sub-sequences q^j are formed by patterns $x^i \in \mathbb{R}^n$, where n is a dimension of individual patterns. They are therefore elements of $(\omega_1)^n$ – dimensional space, where ω_1 is a number of patterns formed from ω_0 sequences in each dimension (typically $\omega_1=4$ for $\omega_0=3$). The clusters of sub-sequences are therefore hyperboxes in $\mathbb{R}^{2(\omega_1)^n}$. Unfortunately, the exponential increase of the dimensionality of the 'shape-space' makes this impractical.

Phase-Space Granulation

We provide here an alternative approach to capturing the nature of sub-sequences of time series that avoids undue augmentation of the input space. The ‘shape’ of sub-sequences is characterized by a range of gradient angles between the first and every other pattern in the sub-sequence (Figure 13). This results in an interval (hyperbox) description of ‘shape’ that is fully compatible with the interval (hyperbox) description of the time series values, as defined in (11) - (13). The advantage of this granulation is that subsequent clustering does not imply any further increase of the input space dimension.

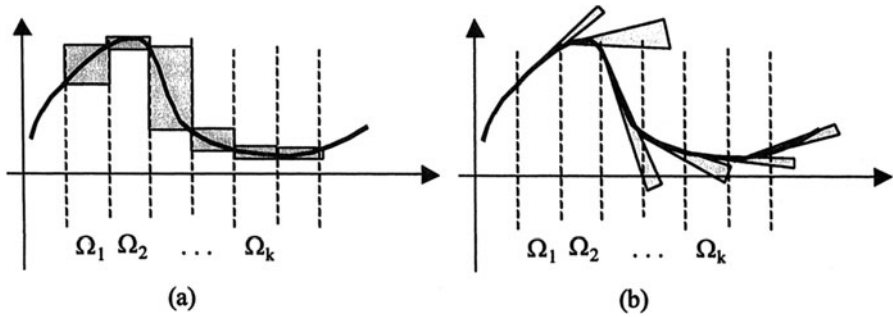


Figure 13. Phase-space granulation: (a) intervals of time series values, (b) intervals of gradient angles (granulation window $\omega_o=3$; see Figure 10).

Since the intervals of time series values in each granulation window are already contained within $[0, 1]$ range only the intervals of gradient angles need to be normalized from $[-\pi/2, \pi/2]$ to $[0, 1]$. We can formalize the phase-space granulation as a mapping of the original data set $\mathbf{X} = \{x^1, x^2, \dots, x^N\}$ onto a set of hyperboxes $\mathbf{H} = \{H^1, H^2, \dots, H^G\}$ where N is a number of elements in the time series and G is the number of granules. A hyperbox H^k is formed as a Cartesian product of two intervals; $H^k = I^k \times J^k$, where I^k is an interval of time series values and J^k is an interval of gradient angles in the k -th granulation window.

$$I^k = \left(\min_{j \in \Omega_k} (x^j), \max_{j \in \Omega_k} (x^j) \right) \quad (14)$$

$$J^k = \left(\min_{j \neq l; j, l \in \Omega_k} (\text{norm}(\text{grad}(x^j, x^l))), \max_{j \neq l; j, l \in \Omega_k} (\text{norm}(\text{grad}(x^j, x^l))) \right) \quad (15)$$

where $\text{grad}(\cdot)$ is an angle-valued gradient function and $\text{norm}(\cdot)$ is a normalization function. The granulation window Ω_k is defined as in (12) and the generalization of the granulation to a multi-dimensional time series is analogous to (13). In this

general case, the dimension of the input space is $4n$ (where n is a dimension of x^k) and the subsequent clustering of hyperboxes H^k does not imply any further increase of the dimension of the pattern space ($H^k \in \mathbb{R}^{4n}$). It is worth emphasizing that increasing the width of the granulation window ω_o reduces the number of granules to N/ω_o while maintaining the dimensionality of the input space. Consequently the computational complexity of the subsequent FCM is reduced by a factor $(\omega_o)^2$. This is in contrast to the granulation proposed in (Das, 1998) where the increase of the width of the granulation window reduces the number of input patterns to N/ω_o , but it increases the dimensionality of the input space by a factor ω_o thus increasing the computational complexity of FCM also by a factor ω_o .

Tests performed on the synthetic time series data indicate that while the inclusion of the gradient of the time series goes some way towards filtering out unrepresentative granules, the FCM clustering of phase-space granulated data is broadly comparable to the results obtained with simple time-domain granulation. Figure 14 provides an illustration of granulation and FCM clustering obtained for $\omega_o=6$ and $\omega_o=8$. Results illustrated in Figure 14(a) are an improvement on the results from Figure 11(d) but there is very little (if any) improvement discernable in Figure 14(b) compared to 11(e).

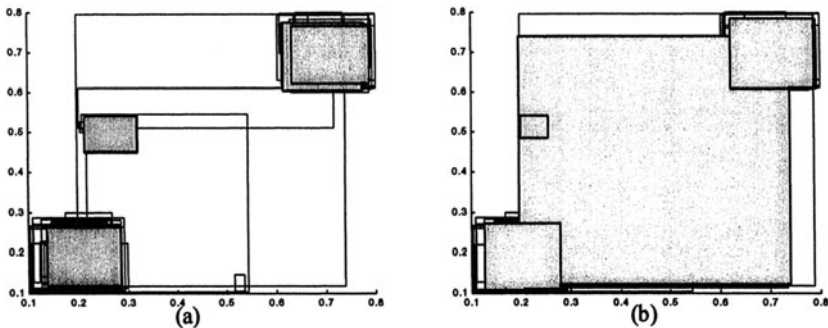


Figure 14. Phase-space granulation and FCM clustering
(a) $\omega_o=6$ (38 granules), (b) $\omega_o=8$ (28 granules).

6. 6 NUMERICAL STUDIES

In this section, we apply granular analysis to a time series of traffic queues collected by an urban traffic control (UTC) system. The data represents traffic on a cross-road during a morning rush-hour. The topology of the selected cross-road is illustrated in Figure 15. The junction is controlled by an adaptive system, called SCOOT (Split-Cycle-Offset Optimization Technique) that attempts to maximize the traffic throughput of the junction by adaptively modifying the duration of the red/green signaling stages. However, the details of SCOOT heuristics, that implement the traffic signals optimization, are not readily available since the system is a

commercial product. This is unfortunate because the development of various high-level traffic management tasks such as in-car traffic information, variable message signs and public transport information (all of which require predictions of traffic flows over extended time-scales) is critically tied to the performance of the SCOOT system (Claramunt et al, 2000; Kosonen et al, 1998). We shall show here that by performing granular analysis of traffic data it is possible to infer operational control rules that can provide a basis for the development of high-level traffic management tasks while, in the same time, leaving SCOOT fully in charge of detailed optimization of traffic signals.

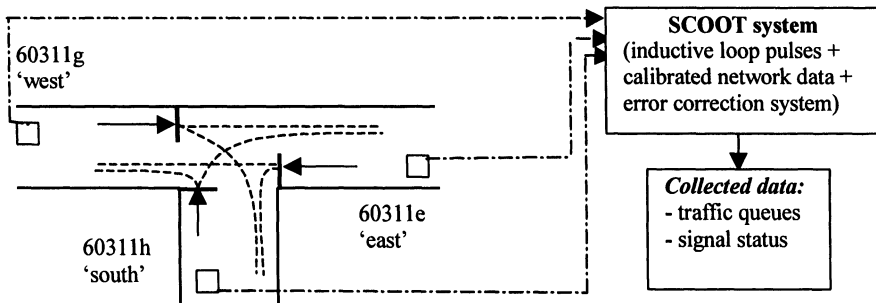


Figure 15. A junction with 3 measured traffic flows (in Mansfield, UK).

While the full Mansfield SCOOT system involves some 40 intersections we limit ourselves, for the sake of clarity of presentation, to just one intersection as illustrated in Figure 15. The three inductive loops are the measuring devices that count discrete pulses generated by cars passing over them. The number of pulses generated by a car is proportional to the length of the car and inversely proportional to its speed. So a small vehicle moving slowly and a large vehicle moving quickly may generate the same number of pulses. This is actually a very advantageous property of this type of measuring devices because it enables focusing on generic rather than specific vehicles. The inductive loop measurements are combined with real-time readings of traffic signal status and also the calibrated travel times between each inductive loop and its corresponding stop-line. On this basis SCOOT is able to estimate the number of vehicles that will arrive at the stop-line during the red signaling stage. This estimate, updated in real-time, is referred to as 'traffic queue measurement'. Since the integration of inductive pulses is prone to systematic error, there are additional inductive loops (not shown on Figure 15), which are used to re-set this error to zero for some specific queue length. In effect, the SCOOT system has a built-in 'safety net' for the traffic queue measurements. By monitoring the 'discharge flows' from the stop-line during the green signaling stage, SCOOT accounts also for the queue remaining from the previous signaling stage in the derived traffic queue measurements.

In the first instance we analyze a 3-dimensional time series of changes of traffic queues in the links links '60311g', '60311e' and '60311h'. We will refer to these links as 'west', 'east' and 'south' respectively. Clearly, the expectation is that the relative changes of traffic queues in any pair of links will reflect the embedded 'rules of operation' of this specific junction.

The original time series consist of 705 discrete measurements for each inductive loop. The readings are time-aligned and form a 3-dimensional vector of system states for 705 time instances. In order to achieve consistent representation of data-points and intervals we increase the dimensionality of the pattern space from 3 to 6 (as illustrated in Figure 8). We apply the granulation and clustering to this 6-dimensional state vector and visualize the results by three 2-D projections. It is worth mentioning that while the granulation and clustering operates on data that is normalized to a unit hypercube ($[0, 1]^3$), the results are converted back to the original data values.

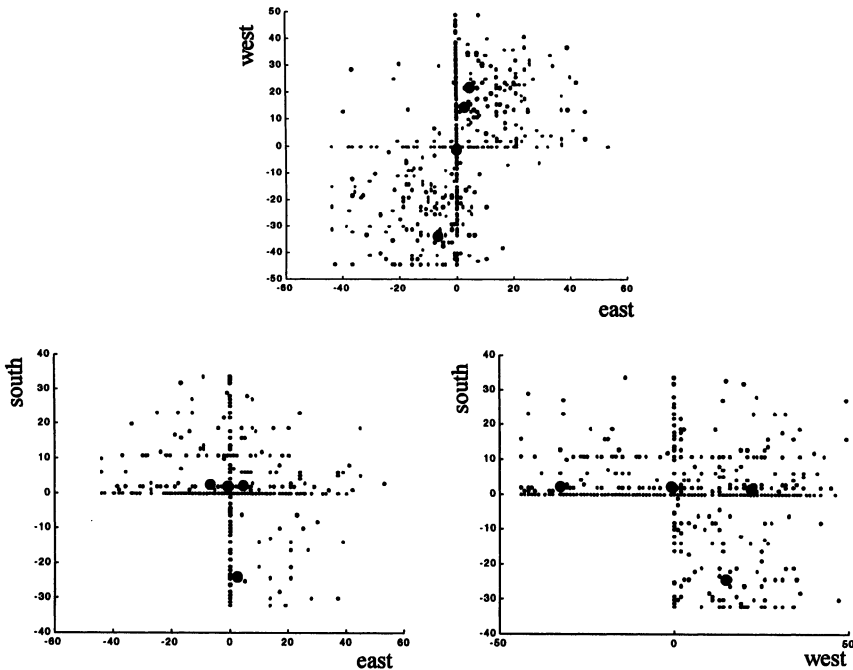


Figure 16. FCM granular prototypes for the level-one granulated data (705 data points).

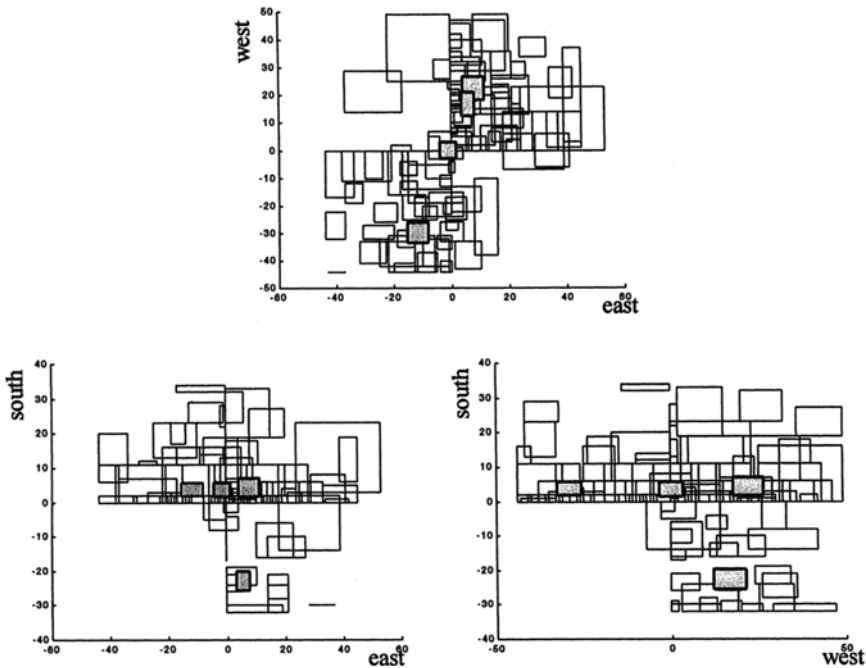


Figure 17. FCM granular prototypes for the level-one granulated data (114 granules).

Figures 16-18 reveal some interesting properties of the system. First, the plot of the original data and FCM prototypes (Figure 16) is somewhat surprising in that significant relationships between traffic in various directions appear not to be fully represented by the prototypes. This is because the data grouped along the axes ('west'=0, 'east'=0 and 'south'=0) exerts undue influence on the FCM algorithm. The situation changes quite dramatically when we consider granulated data (Figures 17-18). In this case, the prototypes cover a larger proportion of data and become more representative of the overall operation of the junction.

Second, the FCM prototypes reveal something that is not obvious from the plot of original data, namely that the queue changes on the 'west' and 'south' link are significantly larger than on the 'east' link. The examination of the physical road layout reveals that the 'south' and 'west' links have separate 'right-turner lanes' and the corresponding inductive loops are spreading there across two rather than just one lane. While the essence of this relationship has been captured by the FCM prototypes built both on original and granulated data, the granular version of FCM appears to deliver more representative results in that the ratio of 'south'/'east' and 'west'/'east' is approximately 2/1 for the granular prototypes and is approximately 4/1 for the point-size prototypes.

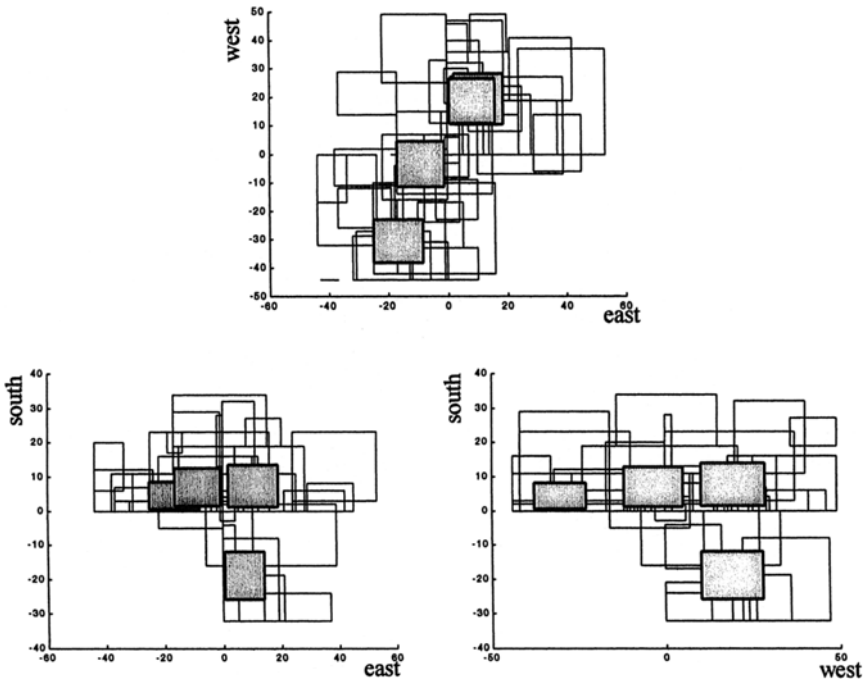


Figure 18. FCM granular prototypes for the level-two granulated data (46 granules).

Third, the granular FCM prototypes, unlike the standard ones, reveal that there is a significant 'right-turner' traffic on the 'west' link. This is represented by a prototype that assumes positive values (queue increases) on the 'west' link when there are negative values (queue decreases) on the 'east' link (compare the 'east'-'west' plots). Notice that there is no similar effect caused by the 'right-turners' on the 'south' link, which means that the operation of the 'south' link is mutually exclusive with 'west' and 'east' links.

Although we have demonstrated, in the previous section, that the time-domain granulation produces much inferior results, we enclose here, for completeness, results obtained for such granulated traffic data. In order to achieve comparability of the results we select $\omega_0=6$, giving 117 granules, which compares to 114 granules from Figure 17. As expected, the time-domain granulation results are poor. Figure 19 shows the FCM prototypes build on time-domain granulated data. The specificity of prototypes is all but lost and while one can discern some similarity in the distribution of prototypes none of the earlier detailed analysis of the operation of the junction seem possible. In fact, the prototypes depicted on the 'south'/'east' and 'south'/'west' projections indicate that it is possible to have simultaneous queue

reduction on the corresponding ‘south’-‘east’ and ‘south’-‘west’ links. This is an erroneous indication since such operation of the junction would clearly lead to a collision and, as such, is specifically prevented by traffic signals. Of course, time-domain granulation can deliver significantly better results if narrower granulation windows are used. However, this defeats the idea of granulation and even with $\omega_0=2$ the results are not as crisply defined as those of Figure 17.

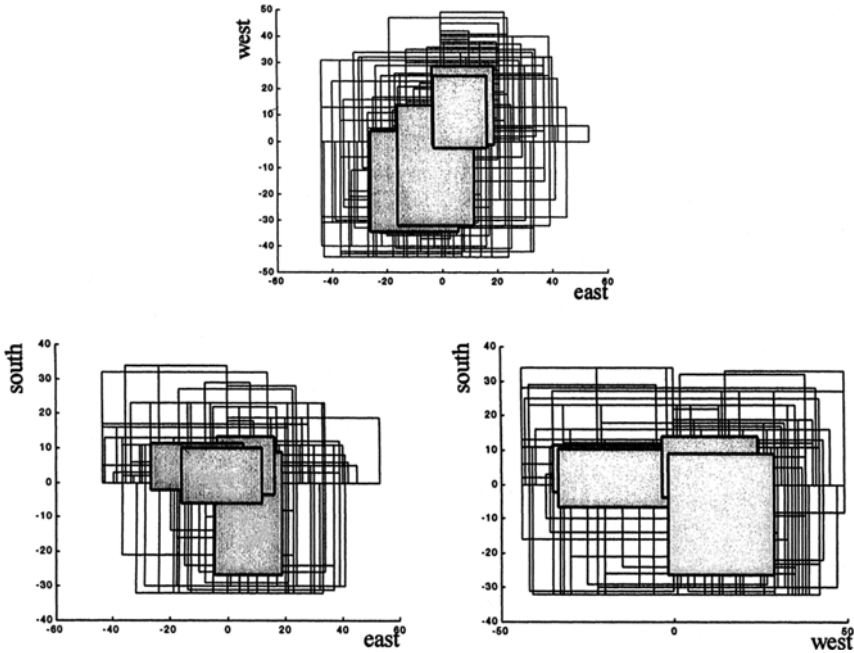


Figure 19. FCM prototypes for the time-domain granulated traffic data ($\omega_0=6$ giving 117 granules).

The application of phase-space granulation to the traffic system data produces similar results to those obtained with simple time-domain granulation (Figure 20). The FCM prototypes build on phase-space granules are significantly less specific than the prototypes obtained with state-space granules and, as such, are less well suited for system modelling purposes. We conclude therefore that the state-space granulation based on maximization of information density has a potential to be of benefit in many practical applications requiring efficient data abstraction.

6. 7 CONCLUSIONS

In this chapter, we discussed a recursive information granulation algorithm that provides a good basis for the iterative refinement of data abstractions. The granulation is based on maximization of information density and it results in easily interpretable granules (hyperboxes). These, in contrast to numeric prototypes, are more user-oriented and deliver a compact characterization of the main relationships existing in the data.

Recursive information granulation exhibits several essential features:

- it helps concentrate on a certain level of detail while ignoring (on purpose) more detailed relationships that may be pertinent only to the higher level of granularity,
- it allows to emphasize the essence of granulation (for instance, monotonic segments of data, segments of equal width, etc.),
- it provides a natural vehicle for converting the problem into a hierarchy of manageable sub-problems. Large, less specific information granules form a first level of analysis that could be afterwards refined by defining more specific information granules capturing more details and geared toward some specific analysis. Formally speaking, denoting an information granule at the higher, and more abstract level, by A , the more detailed analysis relies on information granules B_1, B_2, \dots, B_c such that all of them are included in A , $B_i \subset A$ and

they "cover" A in the sense that $A = \bigcup_{i=1}^c B_i$

While the experimental part of the study concentrated on multi-dimensional time series, the same methodology applies to other multi-dimensional data such as images. The quality of summarization of information granules has been assessed through FCM clustering. Overall the granular description is very much intuitive and qualitative and provides the designer/user with a general insight into the very nature of the phenomenon manifesting through this time series. In this sense, this analysis concurs with a general agenda of qualitative modelling (Sowa, 2000) and fuzzy qualitative modelling (Herera, Martinez, 2001).

REFERENCES

- Bargiela, A. (2001), *Interval and Ellipsoidal Uncertainty Models*, in W. Pedrycz (ed.) *Granular Computing*, Springer Verlag, 23-57.
- Bargiela, A., Pedrycz, W. (2002), Recursive information granulation: Aggregation and interpretation issues, *IEEE Trans. on Syst. Man and Cybernetics*, to appear.
- Berthold, M., Huber, K.-P. (1999), Constructing fuzzy graphs from examples, *Intelligent Data Analysis*, 3(1), pp.37-54.

- Bezdek, J.C. (1981), *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, N. York.
- Box, G.E., Jenkins, G.M. (1970), *Time Series Analysis: Forecasting and Control*, Holden Day, San Francisco.
- Chiu, S. (1996), Method and software for extracting fuzzy classification rules by subtractive clustering, *NAFIPS*, 1996, pp. 461-465.
- Cios, K., Pedrycz, W., Swiniarski, R. (1998), *Data Mining Techniques*, Kluwer Academic Publishers, Boston.
- Claramunt, C., Jiang, B., Bargiela, A. (2000), A new framework for the integration, analysis and visualization of urban traffic data within geographic information systems, *Transportation Research – Part C*, 167-184.
- Das, et al., (1998), Rule discovery from time series, *Proc. of 4th Int. Conf. on Knowledge Discovery and Data Mining*, 16-22.
- Davis, E. (1987), Constraint propagation with interval labels, *Artificial Intelligence*, 24, 347-410.
- Everitt, B.S. (1974), *Cluster Analysis*, Heinemann, Berlin.
- Gabrys, B., Bargiela, A. (2000), General fuzzy min-max neural network for clustering and classification, *IEEE Trans. on Neural Networks*, vol.11, no.3, 769-783.
- Hata, Y., Mukaidono, M. (1999), On some classes of fuzzy information granularity and their representations, *ISMVL'99*, Japan, 288-293.
- Herera, F., Martinez, L. (2001), A model based on linguistic 2-tuples for dealing with multigranular hierarchical linguistic contexts in multi-expert decision making, *IEEE Trans. on Systems Man and Cybernetics*, SMC-B, vol. 31, 2, 227-234.
- Hoppner, F., Klawonn, F., Kruse, R., Runkler, T. (1999), *Fuzzy Cluster Analysis*, J. Wiley, Chichester.
- Huber, P.J. (1981), *Robust Statistics*, J. Wiley, New York.
- Kandel, A. (1986), *Fuzzy Mathematical Techniques with Applications*, Addison-Wesley, Reading, MA.
- Kenneth, D.L., Jeffrey, L.A. (1990), *Robust Regression. Analysis and Applications*, Marcel Dekker, New York.
- Kosonen, I., Bargiela, A., Claramunt, C. (1998), A distributed information system for traffic control, *Proc. 10th European Simulation Symposium (ESS)*, 355-361.
- Kosonen, I., Bargiela, A. (2001), Real-time environment for micro-simulation of urban traffic, *European simulation Symposium ESS'2001*, Marseille, Oct. 2001, 382-386.
- Kuipers, B.J. (1984), *Qualitative Reasoning*, MIT Press, Cambridge, MA.
- Madisetti, V.K., Williams, D.B. (1998), *The Digital Signal Processing Handbook*, CRC Press/IEEE Press, Boca Raton.

- Mani, I., Maybury M.T. (eds.) (1999), *Advances in Automatic Text Summarization*, MIT Press, Cambridge, MA.
- Moore, R.E. (1966), *Interval Analysis*, Prentice Hall, Englewood Cliffs, NJ.
- Moore, R.E. (ed.) (1988), *Reliability in Computing: The Role of Interval Methods*, Academic Press, N. York.
- Pawlak, Z., (1991), *Rough Sets: Theoretical Aspects of Reasoning about Data*, Kluwer Academic, Dordrecht.
- Oppenheim, A.V., Wilsky, A.S. (1983), *Signals and Systems*, Prentice Hall, Englewood Cliffs, NJ.
- Oppenheim, A.V., Schaffer, R.W. (1989), *Discrete-Time Signal Processing*, Englewood Cliffs.
- Pedrycz, W. (1997), *Computational Intelligence: An Introduction*, CRC Press, Boca Raton.
- Pedrycz, W., Gomide, F. (1998), *An Introduction to Fuzzy Sets*, Cambridge, MIT Press, Cambridge, MA.
- Pedrycz, W. (2001), Fuzzy equalization in the construction of fuzzy sets, *Fuzzy Sets and Systems*, 119(2), 321-327.
- Pedrycz, W., Bargiela, A. (2001), Information granulation: A search for data structures, *Knowledge-based Engineering Systems KES 2001*, Osaka, October 2001, 1147-1151.
- Sowa, J.F. (2000), *Knowledge Representation. Logical, Philosophical, and Computational Foundations*, Brooks/Cole, Pacific Grove.
- Thiele, H. (2000), On algebraic foundations of information granulation III, Investigating the HATA-MUKAIDONO approach, *ISMVL 2000*, USA, 2000.
- Zadeh, L.A. (1979), Fuzzy sets and information granularity, In: M.M. Gupta, R.K. Ragade, R.R. Yager, eds., *Advances in Fuzzy Set Theory and Applications*, North Holland, Amsterdam, 3-18.
- Zadeh, L.A. (1996), Fuzzy logic = Computing with words, *IEEE Trans. on Fuzzy Systems*, vol. 4, 2, 103-111.
- Zadeh, L.A. (1997), Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, *Fuzzy Sets and Systems*, 90, 111-117.
- Zadeh, L.A. (1999), From computing with numbers to computing with words-from manipulation of measurements to manipulation of perceptions, *IEEE Trans. on Circuits and Systems*, 45, 105-119.
- Zadeh, L.A., Kacprzyk, J. (1999), *Computing with words in Information/Intelligent Systems*, Studies in Fuzziness and Soft Computing series, Vol. 33 & 34, Physica-Verlag.
- Zimmermann, H.J. (1985), *Fuzzy Set Theory and Its Applications*, Kluwer Academic Publishers, Dordrecht.

GRANULAR PROTOTYPING IN FUZZY CLUSTERING

In the previous two chapters we discussed granulation algorithms that are based on direct manipulation of information granules. Here, we introduce an algorithm for deriving information granules as prototypes in the *clustering process*. The way of revealing a structure in data is realized by maximizing a certain performance index (objective function) that takes into consideration an overall level of matching (to be maximized) and a similarity level between the prototypes (the component to be minimized). It is shown how the relevance of the prototypes translates into their granularity. The clustering method helps identify and quantify anisotropy of the feature space. We also show how each prototype is equipped with its own weight vector describing the anisotropy property and thus implying some ranking of the features in the data space.

8. 1 INTRODUCTION

There is a wealth of clustering techniques (Andenberg, 1973; Bezdek, 1981; Pedrycz, 1997; Zadeh, 1979) and a diversity of ways in which clustering is used in fuzzy modeling and pattern recognition, cf. (Kandel, 1986; Zadeh, 1979; 1996). Clusters are information granules and as such start playing a central role at the algorithmic layer of granular computing. There have been several pursuits along these lines (Gabrys, Bargiela, 2000; Pedrycz, Bargiela, 2002; Simpson, 1992; 1993) yet the area is still in its early development stage. This chapter discusses a logic-driven clustering culminating in a granular type of prototypes. There are several objectives we would like to formulate in this context. First, we would like to build prototypes in a sequential manner so that they could be ranked with respect to their relevance. Second, we would like the clustering algorithm to exhibit significant explorative capabilities. This will be instilled by defining a suitable performance index (objective function). Thirdly, the way in which the prototypes are formed should lend itself to their granular extension.

We address these objectives in a top-down presentation by first discussing the essence of the method and then elaborating on all pertinent details. The

experimental part of the study consists of low dimensional (mainly two-dimensional) patterns, as our intent is to illustrate the efficacies of the proposed clustering and granulation mechanisms. We contrast the algorithm with the Fuzzy C-Means (FCM) being treated as a de-facto standard in fuzzy clustering.

8. 2 PROBLEM FORMULATION

Without the loss of generality we confine our discussion to data (patterns) distributed in an n -dimensional $[0,1]$ hypercube. In what follows, we will be treating the data as points in $[0,1]^n$, say $x \in [0,1]^n$. In general we are concerned with N data points x_1, x_2, \dots, x_N . The “standard” objective of the clustering method (no matter what is its realization) is to reveal a structure in the data set and to present it in a readable and easily comprehensible format. In general, we consider a collection of prototypes to be a tangible and compact reflection of the overall structure. In the approach undertaken here we adhere to the same principle. The prototypes representing each cluster are selected as some elements of the data set. Their selection is realized in such a way that they (a) match (represent) the data to the highest extent while (b) being evidently distinct from each other. These two requirements are represented in the objective function guiding the clustering process. In the sequel we define the detailed components of the optimization. Since the elements in the unit hypercube can be viewed as fuzzy sets, we can take advantage of well-known logic operations developed in this domain. The notion of similarity (equality) between membership grades plays a pivotal role and this concept is crucial to the development of the clustering mechanisms.

Expressing Similarity Between Two Fuzzy Sets

The measure of similarity between two fuzzy sets (in this case a datum and a prototype) $x = [x_1 \ x_2 \dots \ x_n]^T$ and $v = [v_1 \ v_2 \dots \ v_n]^T$ is defined by incorporating the operation of matching (\equiv) encountered in fuzzy sets. The following definition will be used

$$\text{sim}(x, v) = T_{i=1}^n (w_i^2 s(x_i \equiv v_i)) \quad (1)$$

In the above, $T(\cdot)$ and $s(\cdot)$ denote a t -norm and s -norm, respectively. The weights (w_i) quantify an impact of each coordinate of the feature space $[0,1]^n$ on the final value of the similarity index $\text{sim}(\cdot)$. When convenient, we will be using a notation $\text{sim}(x, v; w)$ to emphasize the role played by the weight vector. The similarity between two membership grades is rooted in the fundamental concept of similarity (or equivalence) of two fuzzy sets (or sets). Given two membership grades a and b , (the values of a and b are confined to the unit interval), a similarity level $a \equiv b$ is computed in the form

$$a \equiv b = (a \rightarrow b)t(b \rightarrow a) \quad (2)$$

where the implication operation (\rightarrow) is defined as a residuation (ϕ -operator) (Bouchon-Meunier et al, 1996; Di Nola et al, 1989) that is

$$a \rightarrow b = \sup\{c \in [0,1] \mid atc \leq b\} \quad (3)$$

The above expression of the residuation is induced by a certain t-norm. The implication models a property of inclusion; referring to (3) we note that it just quantifies a degree to which a is *included* in b . The *and* connective used in (2) translates it into a verbal expression

$$(a \text{ is included in } b) \text{ and } (b \text{ is included in } a) \quad (4)$$

which in essence quantifies an extent to which two membership grades are equal. As a matter of fact, the origin of this definition traces back to what we know well in set theory: we say two sets A and B are equal if A is included in B and B is included in A . The visualization of the similarity treated as a function of " a " with " b " regarded as a parameter of this index is included in Figure 1. As expected, it attains 1 if and only if a is equal to b . The function decreases when moving away from " b ". It is however quite asymmetric where this asymmetry arises as a consequence of the implication operations being used in the definition. Note also that the change in the t-norm in the basic definition (2) does not affect the form of the similarity index.

The similarity index is affected by the residuation operation (being more precise, a specific t-norm being used to induce it). For example, Lukasiewicz implication (induced by the Lukasiewicz t-norm) produces a series of piecewise linear plots, Figure 2.

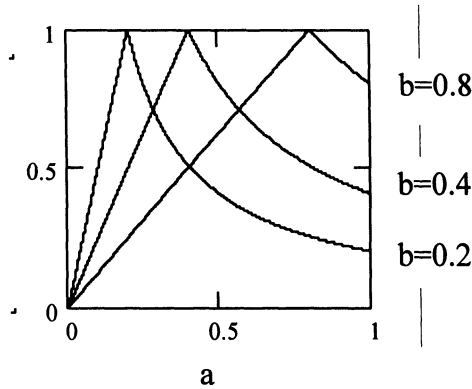


Figure 1. The similarity index $a \equiv b$ regarded as a function of a for selected values of b ; the residuation is induced by the product operation, $a \rightarrow b = \min(1, b/a)$

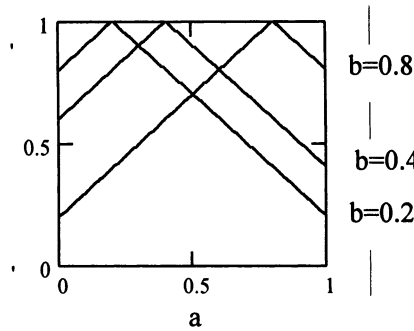


Figure 2. The similarity index $a \equiv b$ regarded as a function of “a” for selected values of “b” and Lukasiewicz implication operation, $a \rightarrow b = \min(1, 1-a+b)$

For some alternative definitions of similarity measures refer to (Ni Nola et al, 1989).

The illustration of the similarity index in case of two variables ($n=2$) is shown in Figure 3. The intent is to visualize the impact of the weights on the performance of the index. It is apparent that high values of the weight reduce the impact of the corresponding variable.

Performance Index (objective function)

Performance index reflects the character of the underlying clustering philosophy. In this work we have adopted performance index that can be concisely described in the following manner. A prototype of the first cluster v_1 is selected as one of the elements of the data set ($v_1 = x_j$ for some $j=1,2,\dots,N$) so that it maximizes the sum of the similarity measures of the form

$$\sum_{k=1}^N \text{sim}(x_k, v_1; w_1) \Rightarrow \text{Max}_{v_1, w_1} \quad (5)$$

with $\text{sim}(x_k, v_1, w_1)$ defined by (1). Once the first cluster (prototype) has been determined (through a direct search across the data space with a fixed weight vector and subsequent optimization of the weights treated as another part of the optimization process), we move on to the next cluster (prototype) v_2 and repeat the cycle. The form of the objective function remains the same throughout the iterative process but we combine now the maximization of the sum of similarity measures (5) with a constraint on the relative positioning of the new prototype. The point is that we want this new prototype, say v_2 , not to “duplicate” the first prototype by being too close to it and thus not representing any new part of the data. To avoid this effect, we now consider the expression of the form

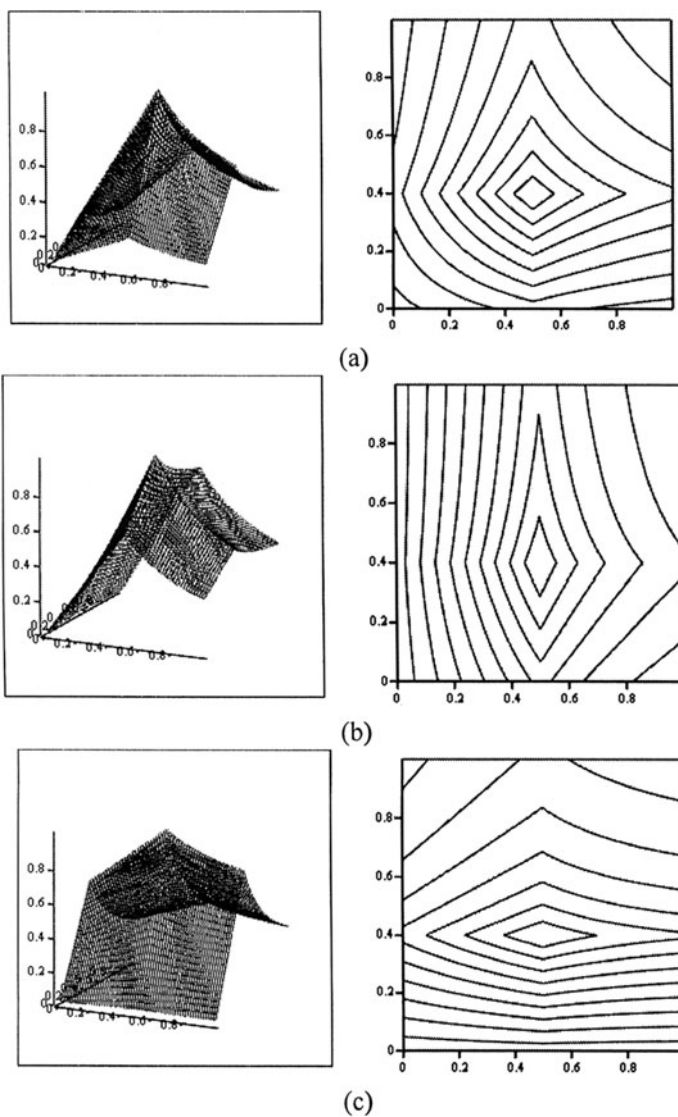


Figure 3. Similarity index (3D plot and two-dimensional contours) for selected values of weight factors: (a) $w_1=0.5$, $w_2=0.5$, (b) $w_1=0.2$, $w_2=0.8$, (c) $w_1=0.8$, $w_2=0.2$. In all cases $v=[0.5, 0.4]$

$$(1 - \text{sim}(\mathbf{v}_2, \mathbf{v}_1; \mathbf{0})) \sum_{k=1}^N \text{sim}(\mathbf{x}_k, \mathbf{v}_2; \mathbf{w}_2) \quad (6)$$

where the first factor $1 - \text{sim}(\mathbf{v}_1, \mathbf{v}_2; \mathbf{0})$ expresses the requirement of \mathbf{v}_2 to be as far apart from \mathbf{v}_1 as possible. The above expression has to be maximized with respect to \mathbf{v}_2 and this optimization has to be carried out with the weight vector (\mathbf{w}_2) involved. In the sequel, we proceed with the determination of the third prototype \mathbf{v}_3 , etc. In general, the optimization of the L -th prototype follows the expression

$$Q(L) = (1 - \text{sim}(\mathbf{v}_L, \mathbf{v}_{L-1}; \mathbf{0}))(1 - \text{sim}(\mathbf{v}_L, \mathbf{v}_{L-2}; \mathbf{0})) \dots (1 - \text{sim}(\mathbf{v}_L, \mathbf{v}_1; \mathbf{0})) \sum_{k=1}^N \text{sim}(\mathbf{x}_k, \mathbf{v}_L; \mathbf{w}_L) \quad (7)$$

As noted, this expression takes into account all previous prototypes when looking for the current prototype. Interestingly, the performance index to be maximized is a decreasing function of the prototype index, that is $L_1 < L_2$ implies that $Q(L_1) \leq Q(L_2)$.

Another observation of interest is that the first prototype constitutes the best representative of the overall data set. Subsequent prototypes are, in effect, the best representatives for the more detailed partitions of data. We can now proceed with the optimization of the weight vector associated with the prototypes.

8. 3 PROTOTYPE OPTIMIZATION

Let us concentrate on the optimization of the performance index in its general form given by (7). Apparently the optimization consists of two phases, that is (a) the determination of the prototype (\mathbf{v}_L) and the optimization of the weight vector (\mathbf{w}_L). These two phases are intertwined yet they exhibit a different character. The prototype is about enumeration out of a finite number of options (patterns in the data set). The weight optimization has not been formulated in detail and now requires a prudent formulation as a constraint type of optimization (without any constraint the task may return a trivial solution). Referring to (7) we observe that it can be written down in the form

$$Q(L) = G \sum_{k=1}^N \text{sim}(\mathbf{x}_k, \mathbf{v}_L; \mathbf{w}_L) \quad (8)$$

Note that the first part of the original expression does not depend on \mathbf{w}_L and can be treated as constant with this regard,

$$G = (1 - \text{sim}(\mathbf{v}_L, \mathbf{v}_{L-1}; \mathbf{0}))(1 - \text{sim}(\mathbf{v}_L, \mathbf{v}_{L-2}; \mathbf{0})) \dots (1 - \text{sim}(\mathbf{v}_L, \mathbf{v}_1; \mathbf{0})) \quad (9)$$

We impose the following constraint on w_L requesting that its components are located in the unit interval and sum up to 1,

$$\sum_{j=1}^n w_{Lj} = 1 \quad (10)$$

The optimization of (8) with respect to w_L for a fixed prototype v_L is expressed as

$$\begin{aligned} \max Q(L) &= G \sum_{k=1}^N \text{sim}(x_k, v_L; w_L) \\ \text{subject to } \sum_{j=1}^n w_{Lj} &= 1 \end{aligned} \quad (11)$$

The detailed derivations of the weight vector is done through the technique of Lagrange multipliers. First, we form an augmented form of the performance index

$$V = G \sum_{k=1}^N \left\{ \sum_{j=1}^n (w_j^2 s(x_{kj} \equiv v_{Lj})) \right\} - \lambda \left(\sum_{j=1}^n w_{Lj} - 1 \right) \quad (12)$$

To shorten the expression, we introduce the notation $u_{ks} = x_{ks} \equiv v_{Ls}$. The derivative of V taken with respect to w_{Ls} (the s -th coordinate of the weight vector) is set to zero and the solution of the resulting equations gives rise to the optimal weight vector

$$\frac{dV}{dw_{Ls}} = 0 \quad \frac{dV}{d\lambda} = 0 \quad (13)$$

The derivatives can be computed once we specify t - and s -norms. For the sake of further derivations (and ensuing experiments), we consider a product and probabilistic sum as the corresponding models of these operations. Furthermore we introduce the abbreviated notation $u_{ks} = x_{ks} \equiv v_s$ for $k=1, 2, \dots, N$ and $s=1, 2, \dots, n$. Taking all of these into account, we have

$$\frac{dV}{dw_s} = G \sum_{k=1}^N \frac{d}{dw_s} \{ A_{ks} w_s^2 s u_{ks} \} - \lambda = 0 \quad (14)$$

where

$$A_{ks} = \sum_{\substack{j=1 \\ j \neq s}}^n (w_j^2 s u_{ks})$$

The use of the probabilistic sum (s -norm) in (14) leads to the expression

$$\frac{d}{dw_s} \{A_{ks} w_s^2 u_{ks}\} = A_{ks} \frac{d}{dw_s} (w_s^2 + u_{sk} - w_s^2 u_{sk}) = 2A_{ks} w_s (1 - u_{ks}) \quad (15)$$

and, in the sequel

$$\frac{dV}{dw_s} = 2Gw_s \sum_{k=1}^N A_{ks} (1 - u_{ks}) - \lambda = 0 \quad (16)$$

From (16) we have

$$w_s = \frac{\lambda}{2G \sum_{k=1}^N A_{ks} (1 - u_{ks})} \quad (17)$$

The form of the constraint, $\sum_{j=1}^c w_j = 1$, produces the following expression

$$\frac{\lambda}{2} \sum_{j=1}^c \frac{1}{G \sum_{k=1}^N A_{kj} (1 - u_{kj})} = 1 \quad (18)$$

or

$$\frac{\lambda}{2} = \frac{1}{\sum_{j=1}^c \frac{1}{G \sum_{k=1}^N A_{kj} (1 - u_{kj})}} \quad (19)$$

Finally inserting (19) into (17) the s-th coordinate of the weight vector reads as

$$w_s = \frac{1}{\frac{\sum_{k=1}^N A_{ks} (1 - u_{ks})}{\sum_{j=1}^c \frac{1}{G \sum_{k=1}^N A_{kj} (1 - u_{kj})}}} \quad (20)$$

Summarizing the algorithm, it essentially consists of two steps. We try all patterns as a potential prototype, for each choice optimize the weights and find a maximal value of $Q(L)$ out of N options available. The one that maximizes this performance index is treated as a prototype. It comes with an optimal weight vector w_L . Each prototype comes with its own weight vector that may vary from prototype to

prototype. Bearing in mind the interpretation of these vectors we can say that they articulate the “local” characteristics of the feature space of the patterns. As seen in Figure 3, the lower the value of the weight for a certain feature (variable), the more essential the corresponding feature is. Importantly, the importance of the features is not the same across the entire space. The space becomes highly anisotropic where prototypes come equipped with different ranking of the features, see Figure 4.

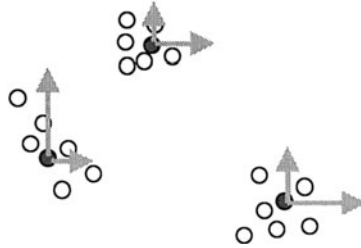


Figure 4. Anisotropy of the feature space of patterns represented by weight vectors associated with prototypes

We discuss a number of low-dimensional synthetic data sets that help us grasp the meaning of the resulting prototypes and interpret their weights.

Example 1

The two dimensional data set shown in Figure 5 exhibits several not very strongly delineated clusters.

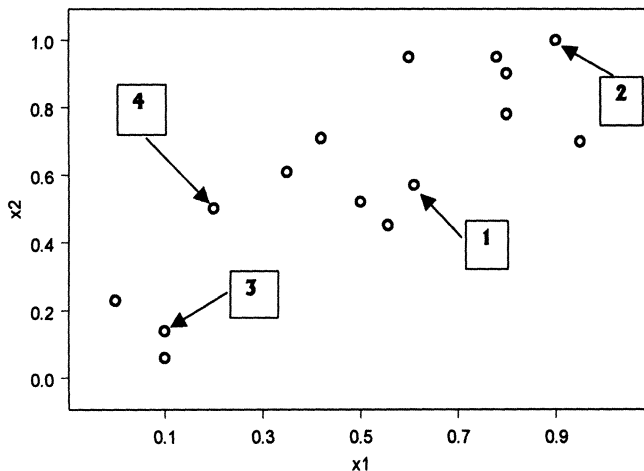


Figure 5. Synthetic data; successive detected prototypes are identified by a corresponding numbers

The clustering is completed out by forming an additional cluster once at a time. The values of the performance index associated with the clusters, a position of the prototypes and their respective weights are summarized in Table 1. As expected, the performance of successive clusters gets lower and the prototypes start locating themselves close to each other. This feature of the clustering approach helps us investigate the relevance of the clusters on the fly and stop the search for more structure once the respective performance indexes start assuming low values. In this example this happens for $c = 5$ at which point the values of the performance index stabilize.

Cluster no.	Prototype	Performance index	w
1	[0.61 0.57]	8.490796	[0.43 0.57]
2	[0.90 1.00]	4.755118	[0.37 0.63]
3	[0.10 0.14]	3.523316	[0.25 0.75]
4	[0.20 0.50]	2.951507	[0.20 0.80]
5	[0.00 0.00]	2.193254	[0.30 0.70]
6	[0.00 0.23]	2.024915	[0.11 0.89]
7	[0.10 0.06]	1.845740	[0.26 0.74]

Table 1. Prototypes and their characterization; the starting point where the values of the performance index stabilize has been highlighted.

Table 1 includes also the weight vectors associated with the prototypes. They reflect upon the “local” properties of the feature space. From their analysis (recall that lower value of the weight means higher relevance of the feature in the neighborhood of the given prototype), we learn that the first feature (x_1) is more relevant than the second one. This is a quantification of the visual inspection: as seen From Figure 5, when projecting the data on x_2 they tend to be more “crowded” (start overlapping) in comparison with their projection on x_1 .

The prototypes produce nonlinear classification boundaries as shown in Figure 6.

For comparative reasons we carried out clustering using FCM; the resulting prototypes and the boundaries between the clusters are included in Figure 7. It can be seen that the nonlinear boundaries between the clusters identified through maximization of the similarity measure afford much more refined partition of the pattern space.

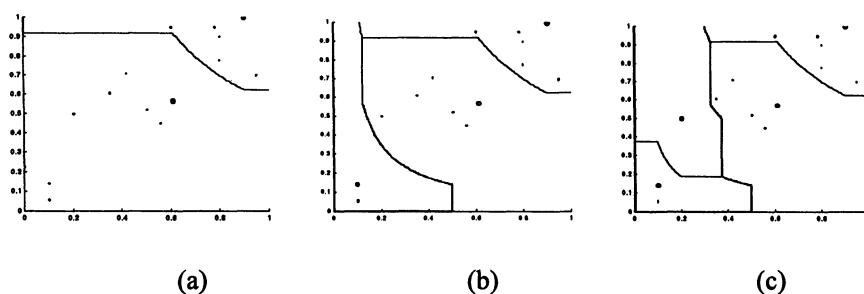


Figure 6. Classification regions for (a) 2 clusters, (b) 3 clusters, and (c) 4 clusters, identified through maximization of the similarity measure.

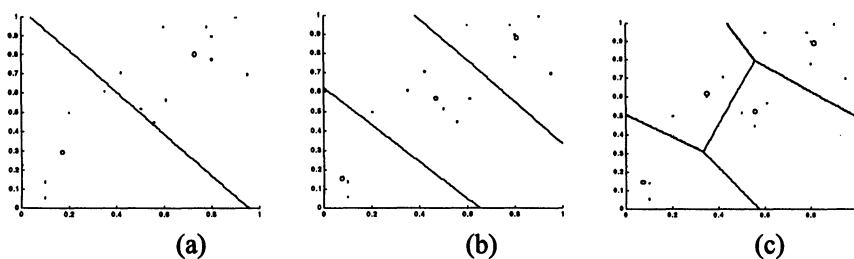


Figure 7. FCM clustering and the implied partition of the pattern space for (a) $c=2$, (b) $c=3$, and (c) $c=4$, clusters.

Example 2. We consider a four-dimensional data set

Pattern no.	Coordinates	Prototype
1	0.80 0.10 0.60 0.30	
2	0.50 0.20 0.40 0.31	
3	0.60 0.30 0.10 0.35	
4	0.40 0.18 0.87 0.40	$\leftarrow 2$
5	0.90 0.15 0.50 0.32	$\leftarrow 4$
6	0.20 0.95 0.65 0.30	
7	0.20 0.40 0.30 0.31	$\leftarrow 3$
8	0.70 0.20 0.63 0.28	$\leftarrow 1$
9	1.00 0.00 1.00 0.31	
10	0.05 0.15 0.42 0.33	

The “optimal” number of clusters is equal to 4 (at this number we see “flattening-out” of the values of the performance index, which means that the maximization of the similarity between data and prototypes is counterbalanced by the increase of similarity between the prototypes), Figure 8. The weight vectors of the prototypes, Table 2, tell an interesting story: the feature space is quite isotropic and in all cases the first feature (x_1) carries a higher level of relevance (the first coordinate of weight vector of each prototype is constantly lower than the other). This is highly intuitive as the patterns are more “distributed” along the first axis (x_1), which makes it more relevant (discriminatory) in this problem.

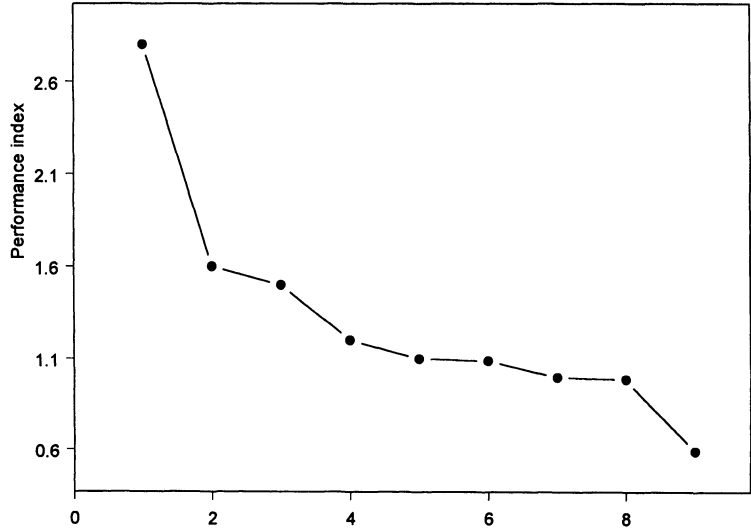


Figure 8. Performance index versus number of clusters (c).

prototype no.	weight vector
1	0.12 0.12 0.16 0.60
2	0.16 0.18 0.14 0.52
3	0.04 0.04 0.06 0.86
4	0.06 0.09 0.15 0.70

Table 2. Weight vectors of the first four prototypes.

Example 3. This two-dimensional data, Figure 10, shows a structure that has three condensed clusters but also includes 2 points that are somewhat apart from the clusters.

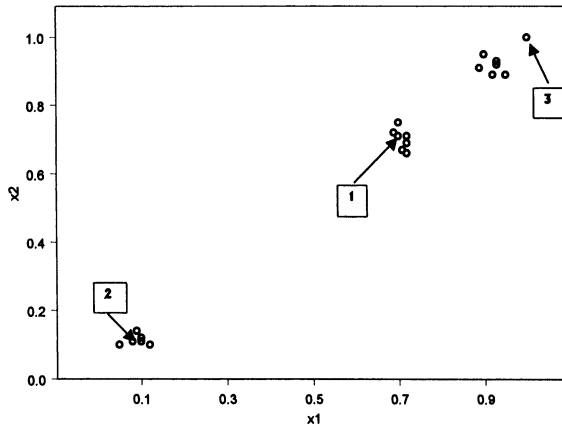


Figure 9. Two-dimensional synthetic data with three first prototypes identified by the clustering algorithm.

The results are included in figure 9. The values of the performance index are visualized in Figure 10. It can be seen that the performance index “flattens-out” for five clusters, which corresponds to identifying significantly distinct data groupings.

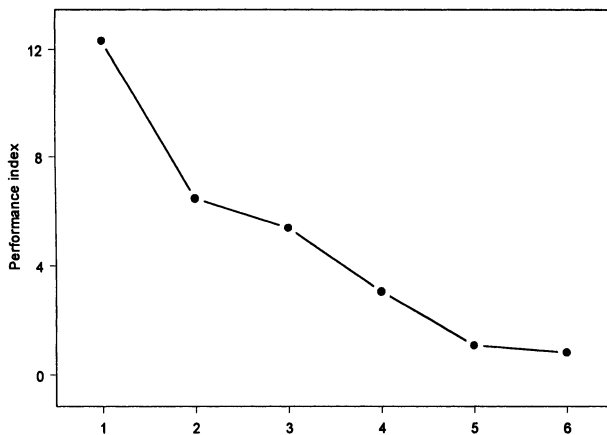


Figure 10. Performance index versus number of clusters (c).

Example 4. This two-dimensional data set reveals two very unbalanced clusters – the first group is evidently dominant (100 patterns) over the second cluster (which consists of 5 data points), Figure 11.

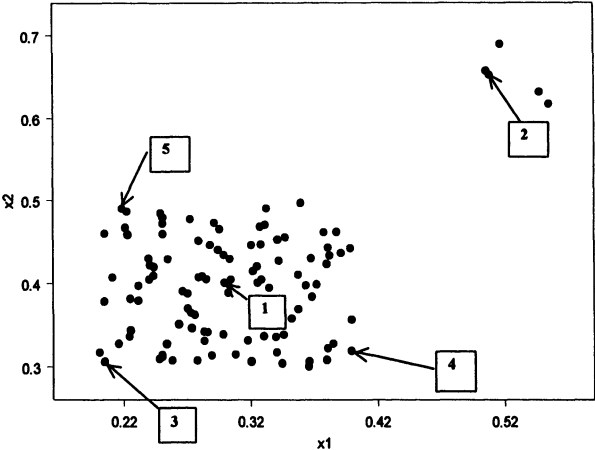


Figure 11. Two-dimensional data set with two unequal clusters; the consecutive prototypes produced by the method were identified by numbers.

As we start building the prototypes, they start representing both clusters in more detail. The second prototype in the sequence has been assigned to the small cluster meaning that the method is after some still not represented parts of the data structure. We may say that the form of the performance index promotes a vigorous exploration of the data space and acts against “crowding” of the clusters in a close vicinity of each other. The consecutive clusters are after the details of the larger cluster as they start unveiling some substructures. Noticeably, the sixth prototype is assigned to the small cluster, Table 3.

Prototype no.	Location	Performance index	Weight vector
1	(0.300100 0.400800)	83.694626	[0.44 0.56]
2	(0.508700 0.652100)	33.989677	[0.48 0.52]
3	(0.205200 0.305700)	26.135773	[0.39 0.61]
4	(0.399500 0.318400)	9.132071	[0.42 0.58]
5	(0.219200 0.489600)	5.200340	[0.42 0.58]
6	(0.555300 0.617200)	1.585717	[0.41 0.59]
7	(0.359900 0.496900)	0.598020	[0.51 0.48]

Table 3. Prototypes of the clusters, their performance index and weight vectors. The shadowed row highlights a sharp drop in the values of the performance index.

It is instructive to compare these results with the structure revealed by the FCM (Figure 12). As anticipated (and this point was raised in the literature), FCM ignores the smaller cluster and it becomes primarily focused on the larger cluster. Only with the increase of the number of the clusters we start capturing the smaller of the clusters yet it happens later than we have reported in the previous method.

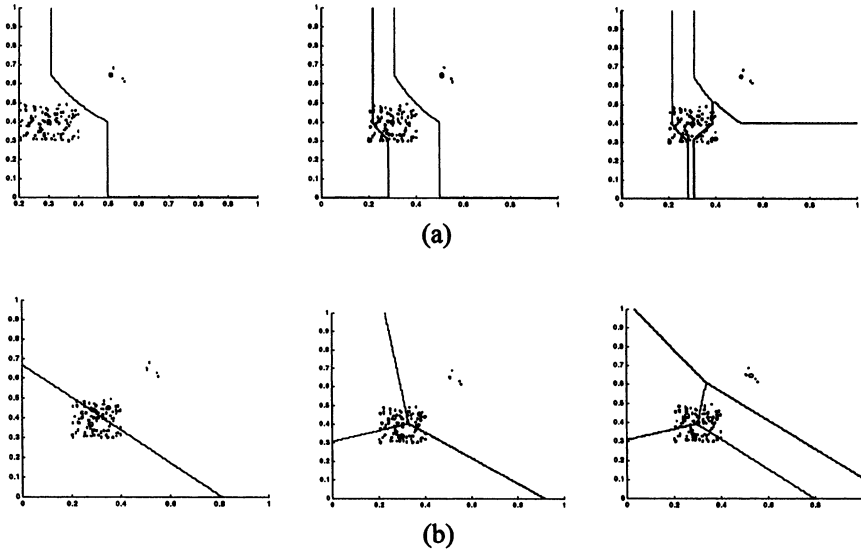


Figure 12. Partition of the pattern space implied by: (a) the similarity measure based clustering, and (b) the FCM clustering; for 2,3, and 4 clusters.

Example 5. The glass data set comes from the repository of Machine Learning (<http://www.ics.uci.edu/~mllearn/MLRepository.html>) and concerns classification of several categories of glass. The study was motivated by criminological investigations. There are 9 attributes (features) that are used in the classification, e.g., refractive index and a content of iron, magnesium, aluminum, etc. in the samples. There are seven classes (categories) identified in the problem.

In the experiment we use first 100 patterns. The performance index for the individual prototypes is shown in Figure 13. The plausible number of clusters is 5 since the performance index again “flattens-out” for larger number of clusters.

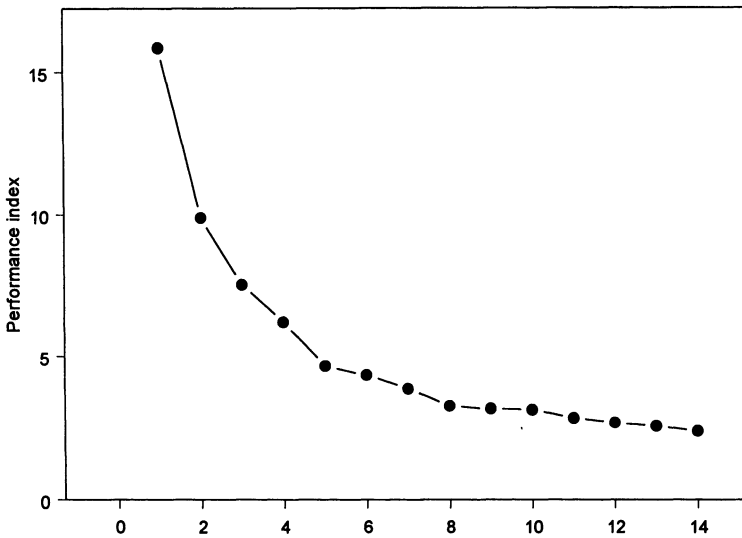


Figure 13. Performance index versus number of clusters.

As the weight vectors of the individual prototypes are concerned (we confine ourselves to 5 most dominant prototypes) they show some level of anisotropy with the features being ranked quite consistently in the context of the individual prototypes. The mean values and standard deviations of the weights of the first five prototypes are shown below

mean values

0.05975 0.02559 0.05539 0.06384 0.05388 0.07063 0.04202 0.61093 0.01828

standard deviations

0.02700 0.01306 0.02453 0.03086 0.02068 0.03929 0.02470 0.16237 0.00771

The feature no. 8 can be clearly identified as relatively insignificant while the most essential ones are {2, 7, 9}. Their standard deviation is also quite low.

8. 4 THE DEVELOPMENT OF GRANULAR PROTOTYPES

The inherently logic nature of the clustering technique lends itself to the development of prototypes that are represented as Cartesian products of intervals in the feature space. Our anticipation is that the granularity of the prototypes gives us a better insight into the nature of the data as well as the relevance of the prototype

itself. The formal framework of building granular prototypes can be introduced as follows. Consider v to be the prototype $v \in [0,1]^n$ already determined in the way discussed in Section 3. It comes with its weight vector w . We can compute an average of similarity (q) between this prototype and all patterns by taking the following sum

$$q = \frac{1}{N} \sum_{k=1}^N \text{sim}(x_k, v; w) \quad (21)$$

(note that (21) is analogous to (8) with an exception that we do not consider here an interaction of v with other prototypes and that we normalize the result). This average similarity serves as a useful indicator of the relevance of the prototype. Now let us determine such values of u_i for which (21) holds. As u_i is effectively a similarity level between x_i and v_i , in essence it implies the interval built around v_i . To see this, note that $u_i = x_i \equiv v_i$ so if v_i and u_i are given, one can determine the range into which x_i should fall in order to satisfy this equality. This range is just an interval (along i -th coordinate) that contains the prototype.

To form the granular prototype the process is repeated for all features, $i=1, 2, \dots, n$, and we formulate and handle explicitly two optimization tasks arising here. The first one concerns the determination of the values of u_i , $i=1, \dots, n$, so that they satisfy (21). The second task is an inverse problem emerging in the setting of the similarity index.

Optimization of the Similarity Levels

As a part of the construction of the granular prototypes we encounter the problem of determining the matching levels along individual features given the weight vector w and the overall matching level q . In other words we are looking for $u = [u_1 \ u_2 \ \dots u_n]$ such that

$$T(w_j^2 s u_j) = \gamma \quad (22)$$

where u collects the matching levels between given prototype v and some other pattern x . The above problem is not trivial and no closed form solution can be derived. Some iterative optimization should be deployed here. Bearing this in mind we reformulate (22) as a standard MSE approximation problem

$$P = [T(w_j^2 s u_j) - \gamma]^2 \rightarrow \text{Min}(u) \quad (23)$$

whose solution is obtained by a series of modifications of u through the gradient-based scheme, namely

$$\mathbf{u}(\text{new}) = \mathbf{u} - \alpha \nabla_{\mathbf{u}} P \quad (24)$$

where α denotes a positive learning rate. The detailed expression for the update can be derived for some predefined form of the triangular norm. Again using the product and probabilistic sum we produce a detailed expression for the gradient,

$$u_k(\text{new}) = u_k - \alpha \frac{\partial P}{\partial u_k} \quad (25)$$

$k=1, 2, \dots, n$. The detailed expression for the derivative is given as

$$\frac{\partial P}{\partial u_k} = 2 \left[T_{j=1}^n (w_j^2 s u_j) - \gamma \right] \frac{\partial}{\partial u_k} (T_{j=1}^n (w_j^2 s u_j)) \quad (26)$$

The inner derivative can be handled for specific t- and s-norm. For a certain pair of them (t-norm: product, s-norm: probabilistic sum), we have

$$\frac{\partial}{\partial u_k} (T_{j=1}^n (w_j^2 s u_j)) = \frac{\partial}{\partial u_k} (B_k (w_k^2 + u_k - w_k^2 u_k)) = B_k (1 - w_k^2)$$

where B_k computes using t-norm when excluding the index of interest (k)

$$B_k = T_{\substack{j=1 \\ j \neq k}}^n (w_j^2 s u_j)$$

An Inverse Similarity Problem

The inverse problem coming with the similarity index can be formulated as follows: given b and γ (both in the unit interval), determine all possible values of “ x ” such that $x \equiv b = \gamma$. The character of the solution can be easily envisioned by augmenting this equality by its graphical interpretation, Figure 14. This figure underlines that the problem being formulated as above requires some refinement in order to enhance the interpretability of the solution and assure that it always exist. This can be done by moving from the equality to the inequality format of the relationship

$$x \equiv b \leq \gamma \quad (26)$$

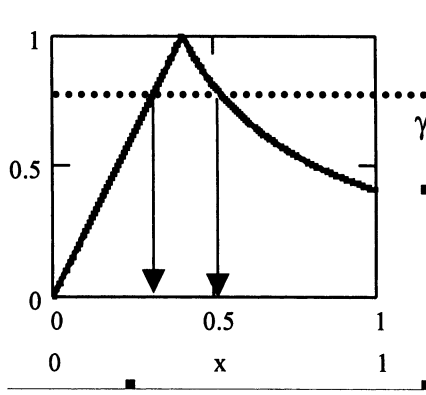


Figure 14. Inverse matching problem: computing an interval of solutions to $x \equiv b \leq \gamma$.

The solution to it arises in a form of a confidence interval (or simply interval) implied by a certain value of γ . This solution (interval) is a manifestation of the granularity of the prototype for a given feature. The solution to (26) can be obtained analytically for a specific type of the t-norm (or implication). As shown in Figure 14, the solution always exists (that is there is always a nonempty interval for any given value of γ). The granularity of the prototype is a monotonic function of γ : higher values of γ imply higher values of granularity, i.e., narrow intervals of the granular prototype. For some critical (low enough) value of γ , the interval expands to the entire unit interval so we have a granular prototype of the lowest possible level of granularity.

Moving on to the detailed calculations, the interval of the granular prototype $[x^-, x^+]$ is equal to

for $x \rightarrow b = \min(1, b/x)$

$$x^- = \gamma b, \quad x^+ = \min(1, b/\gamma) \quad (27)$$

for $x \rightarrow b = \min(1, 1-x+b)$

$$x^- = \max(0, \gamma - 1 + b), \quad x^+ = \min(1, 1 - \gamma + b) \quad (28)$$

(the above expressions are determined by considering the increasing and decreasing portions of the matching index as illustrated in Figure 14.

Continuing the previous examples the resulting granular prototypes are shown in Figure 15 and 16 for Example 1 and 3, respectively. The same granular prototypes summarized as triples of the form {lower_bound, mode, upper_bound} are included in Table 5. Note that by the mode we mean an original numeric value around which the granular prototype is constructed. The optimization of the degrees matching (**u**) was completed by running the gradient based learning with $\alpha=0.05$ for 100 iterations. The initial values of **u**'s are set up as small (near zero) random numbers.

Prototype 1:	{0.431784	0.610000	0.861773 }	{ 0.302657	0.570000	1.000000 }
Prototype 2:	{0.570009	0.900000	1.000000 }	{0.507569	1.000000	1.000000 }
Prototype 3:	{0.035332	0.100000	0.283029 }	{0.015797	0.140000	1.000000 }
Prototype 4:	{0.091757	0.200000	0.435936 }	{0.081955	0.500000	1.000000 }

(a)

Prototype 1:	{0.545043	0.720000	0.951118 }	{0.457757	0.710000	1.000000 }
Prototype 2:	{0.042248	0.100000	0.236699 }	{0.035519	0.120000	0.405423 }
Prototype 3:	{0.784826	1.000000	1.000000 }	{0.477658	1.000000	1.000000 }
Prototype 4:	{0.017296	0.050000	0.144543 }	{0.016742	0.100000	0.597287 }

(b)

Table 5. Granular prototypes represented as triples of lower bounds, modes (numeric values of the prototypes) and upper bounds (a) Example 1 and (b) Example 3.

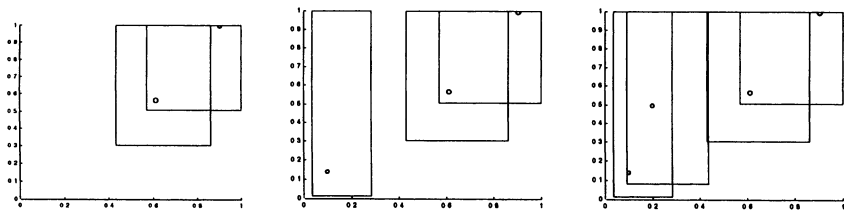


Figure 15. 2- 3- and 4 granular prototypes calculated for data from Example 1.

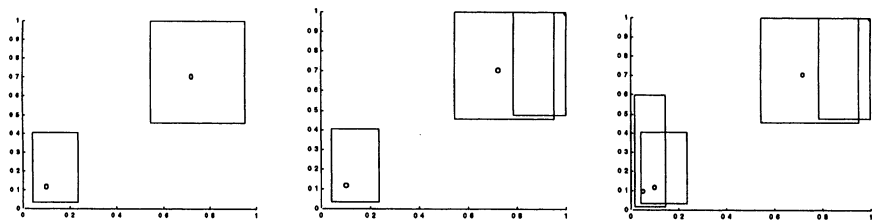


Figure 16. 2- 3- and 4 granular prototypes calculated for data from Example 3.

These granular prototypes reinforce and quantify our perception of structural dependencies in data. In the first case, Figure 15, we note that the first component of the structure resides in the right upper quadrant of the coordinates and this shows very clearly in the distribution of the granules. As a matter of fact prototype 1 and 2 overlap (meaning that there is some redundancy. The next granule (implied by the third cluster) is essential to the quantification of the structure; it occupies the area close to the origin. The fourth cluster overlaps the third one. Noticeably, all granules are elongated along the second variable and this very much quantifies our observation about the limited relevance of this variable (note that all corresponding weights for the second variable are substantially high). The conclusion is that the granules tend to “expand” and occupy the space wherever it is possible; this expansion is visible for x_2 . The granular character of the prototypes in Figure 16 is again a meaningful manifestation of the structure. The two first granules are far apart (and represent the two evidently distinct groups of data). The boxes do not discriminate between the variables viewing them as equally essential. The third granule overlaps with the first one as these two clusters are relatively close. The fourth cluster has a strong resemblance (and overlap) to the second granule.

As this analysis reveals, we can envision a structure of the data by inspecting the resulting granular prototypes. First, these granules help us position clusters in the data space (it is worth stressing that the numeric representation does not support this form of analysis). Second, we can envision a general geometry of data that could be helpful in the design of more detailed classifiers or other models. The granules may also exhibit some level of overlap (no matter how such overlap is expressed in a formal fashion). This may help reason about possible relevance and redundancy of some of these clusters.

8. 5 CONCLUSIONS

We have introduced a new logic-based approach to building information granules. The main and unique features of this approach include:

- *logic-based character of processing*. The search for structure in data is accomplished by exploiting fuzzy set operations. In particular, this concerns the matching operation that is easily interpretable and comes with a well defined semantics
- *sequential construction of the prototypes*. The number of the clusters is not fixed in advance but can be adjusted dynamically depending upon the performance of the already constructed prototypes. The prototypes themselves are constructed starting from the most “significant” (relevant) so that they come ranked
- *identification and quantification of possible anisotropy* of the feature space. The weight vectors coming with the individual prototypes help quantify the importance of the features. The importance of the features can be local and the ranking the features can vary from prototype to prototype

- *development of granular prototypes* realized on a basis of the clustering results. We showed how the relevance of the prototype can be translated into its granular extension

REFERENCES

- Anderberg, M.R. (1973), *Cluster Analysis for Applications*, Academic Press, N. York.
- Bargiela, A. (2001), Interval and ellipsoidal uncertainty models, In: W. Pedrycz (ed.) *Granular Computing*, Physica Verlag, Heidelberg, 23-57.
- Bargiela, A., Pedrycz, W., Hirota, K. (2002), Data granulation through optimization of similarity measure, *Archives of Control Sciences*, to appear.
- Bargiela, A., Pedrycz, W. (2002), A model of granular data: A design problem with the Tschebyshev FCM, *International Journal of Soft Computing*, to appear.
- Bezdek, J.C. (1981), *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, N. York.
- Bouchon-Meunier, B., Rifqi, M., Bothorel, S. (1996), Towards general measures of comparison of objects, *Fuzzy Sets and Systems*, 84, 2, 143-153.
- Di Nola, A., Sessa, S., Pedrycz, W., Sanchez, E. (1989), *Fuzzy Relational Equations and Their Applications in Knowledge Engineering*, Kluwer Academic Press, Dordrecht.
- Delgado, M., Gomez-Skarmeta, F., Martin, F. (1997), A fuzzy clustering-based prototyping for fuzzy rule-based modeling, *IEEE Transactions on Fuzzy Systems*, 5(2), 223-233.
- Duda, R.O., Hart, P.E., Stork, D.G. (2001), *Pattern Classification*, 2nd edition, J. Wiley, NY.
- Gabrys, B., Bargiela, A. (2000), General fuzzy Min-Max neural network for clustering and classification, *IEEE Trans. on Neural Networks*, Vol. 11, No. 3, pp. 769-783.
- Hoppner F. et al (1999), *Fuzzy Cluster Analysis*, J. Wiley, Chichester.
- Ishibuchi, H., Nozaki, K., Yamamoto, N., Tanaka, H. (1995), Selecting fuzzy if-then rules for classification problems using genetic algorithms", *IEEE Trans. on Fuzzy Systems*, (3)3, 260-270.
- Kandel, A. (1986), *Fuzzy Mathematical Techniques with Applications*, Addison-Wesley, MA.
- Pedrycz, W. (1991), Neurocomputations in relational systems, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13, 289-296.
- Pedrycz, W., Rocha, A. (1993), Knowledge-based neural networks, *IEEE Trans. on Fuzzy Systems*, 1, 254-266.
- Pedrycz, W. (1997), *Computational Intelligence: An Introduction*, CRC Press, Boca Raton, FL.
- Pedrycz, W. (1998), Conditional fuzzy clustering in the design of radial basis function neural networks, *IEEE Transactions on Neural Networks*, 9 no. 4, 601- 612.

Pedrycz, W., Vasilakos, A.V. (1999), Linguistic models and linguistic modeling, *IEEE Trans. on Systems Man and Cybernetics*, vol. 29, no. 6, 745-757.

Pedrycz, W., Bargiela, A. (2001), Information granulation: A search for data structures, *Knowledge-based Engineering Systems KES 2001*, Osaka, October 2001, 1147-1151.

Pedrycz, W., Bargiela, A. (2002), Granular Clustering: A granular signature of data, *IEEE Trans. on Systems Man and Cybernetics*, Vol 32, No. 2, 212-224.

Simpson, P.K. (1992), Fuzzy Min-Max neural networks – Part1: Classification, *IEEE Trans. on Neural Networks*, Vol 3, No. 5, pp. 776-86.

Simpson, P.K. (1993), Fuzzy Min-Max neural networks – Part2: Clustering, *IEEE Trans. on Neural Networks*, Vol 4, No. 1, pp. 32-45.

Sudkamp, T. (1993), Similarity, interpolation, and fuzzy rule construction, *Fuzzy Sets and Systems*, vol. 58, no. 1, 73-86.

Zadeh, L.A. (1979), Fuzzy sets and information granularity, In: M.M. Gupta, R.K. Ragade, R.R. Yager, eds., *Advances in Fuzzy Set Theory and Applications*, North Holland, Amsterdam, 3-18.

Zadeh, L.A. (1996), Fuzzy logic = Computing with words, *IEEE Trans. on Fuzzy Systems*, vol. 4, 2, 103-111.

Zadeh, L.A. (1997), Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, *Fuzzy Sets and Systems*, 90, 1997, pp. 111-117.

LOGIC-BASED FUZZY CLUSTERING

This chapter continues with the theme of granulation through logic-based expansion of the standard FCM clustering. The proposed algorithm captures the logic fabric of structure in a data set by describing it in the form of a union of clusters (that is fuzzy relations) determined by the clustering algorithm. In contrast to the standard FCM, the elements (clusters) are combined together as a union of such fuzzy relations. Both clusters and this form of combination arises as a constraint in the clustering method. In this sense, the clustering environment discussed here gives rise to clustering that is regarded as logic-driven data decomposition. A detailed algorithm is presented along with some illustrative examples.

9. 1 INTRODUCTION AND PROBLEM FORMULATION

Fuzzy clustering has been regarded as one among the most popular approaches to data granulation and data analysis, in general (Backer, 1995; Bezdek, 1981). The objective function-based algorithms are an attractive alternative due to the well-known underlying concept, interpretation, computational properties and well-rounded optimization framework of the iterative procedures therein. In spite of the diversity in the formulation of the problem itself, see (Bezdek, 1981; Dave, 1997; Krishnapurnam, 1995), the standard fuzzy partition resulting from the optimization of the objective function satisfies the following constraint of an evident probabilistic character

$$\sum_{i=1}^c u_{ik} = 1 \quad (1)$$

where "k" denotes the k-th pattern in the data set and "i" varies over all groups (clusters). In other words, we require that a sum of the grades of membership is equal to 1. By the same token, a grade of membership indicates a commitment of a given pattern to a certain class and as such carries a meaning of a membership degree. Apparently, the higher the grade of membership associated with this pattern,

the closer its resemblance to the class under discussion. It is interesting to emphasize that while each row of the partition matrix U can be regarded as a fuzzy set (its discrete membership function), the above constraint does not exhibit any logic nature. To alleviate this inconsistency and underline the logic fabric of the clustering, we reformulate the clustering problem (1) as follows. Denote by u_i the i -th row of the partition matrix U . Evidently, u_i is a fuzzy set. These fuzzy sets (consecutive rows of U) constructed through fuzzy clustering are combined in an *or*-like manner meaning that we require that the following logic constraint is satisfied

$$\bigcup_{i=1}^c u_i = 1 \quad (2)$$

The above expression exhibits a clear and well-defined semantics. We refer to the fuzzy clustering exploiting this condition as a logic-based fuzzy clustering. The expression (2) has a direct logic connotation which states that the result of clustering, namely a structure revealed in the data set under discussion, is regarded as a union of the information granules - fuzzy sets identified there,

$$u_1 \text{ or } u_2 \text{ or } \dots \text{ or } u_c \quad (3)$$

Usually in the setting of fuzzy sets, an *or* operation is realized with the aid of *s*-norms, cf. (Butnariu, 1993; Klir, 1995; Pedrycz, 1997; Bargiela, Pedrycz, Hirota, 2002; Zimmermann, 1991). Then (2)-(3) can be written down as

$$\bigvee_{i=1}^c u_i \quad (4)$$

In the case of the maximum operation (*s*-norm) which is in common use, (4) in conjunction with (2) reads as

$$\bigvee_{i=1}^c u_{ik} = \max_i u_{ik} = 1 \quad (5)$$

and can be viewed as a clustering requirement.

It is worth underlining that sometimes the probabilistic-oriented clustering constraint (1) was replaced by a so-called possibilistic clustering condition (Dave, 1997; Krishnapuram, 1995) (under which we require that the membership grades are confined to the unit interval). Nevertheless the resulting clustering algorithm does not lend itself to any logic-based interpretation being similar to the one we are to pursue here.

9.2 THE ALGORITHM

Here, we discuss the details of the clustering algorithm by formulating its underlying optimization problem, solving it and coming up with a complete algorithm. The objective function to be minimized takes on the form

$$Q = \sum_{i=1}^c \sum_{k=1}^N u_{ik}^m \| \mathbf{x}_k - \mathbf{v}_i \|^2 = \sum_{i=1}^c \sum_{k=1}^N u_{ik}^m d_{ik}^2 \quad (6)$$

with the logic-based constraint defined by (2) and is based on the maximum operation. (we will elaborate on the general case involving any s-norm later on as it requires a more comprehensive treatment). $\|\cdot\|$ (that is d_{ik}) denotes a distance function between pattern \mathbf{x}_k and prototype \mathbf{v}_i , $U = [u_{ik}]$ is a partition matrix and "m" is known as a fuzzification factor, $m > 1$.

In other words, the logic-based clustering is concerned with the constraint optimization

$$\min_{\substack{U \in \mathbf{U} \\ \mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c}} Q \quad (7)$$

where \mathbf{U} stands for a family of the fuzzy partitions that is

$$\mathbf{U} = \{U \mid 0 < \sum_{k=1}^N u_{ik} < N \text{ for all } i = 1, 2, \dots, c \text{ and } \bigvee_{i=1}^c u_{ik} = 1 \text{ for all } i = 1, 2, \dots, N\} \quad (8)$$

As usual, the solution to (7) splits into two independent tasks that is (a) a determination of the partition matrix and (b) computing the prototypes of the clusters. The first one takes into account the logic constraints while the second one is constraint-free. Proceeding now with the optimization of the partition matrix we convert the problem into its constraint-free counterpart by applying a technique of Lagrange multipliers. Following this transformation, the augmented objective function V comes in the form

$$V = \sum_{i=1}^c \sum_{k=1}^N u_{ik}^m d_{ik}^2 - \lambda (\bigvee_{i=1}^c u_{ik} - 1) \quad (9)$$

where λ stands for the Lagrange multiplier (note that the problem concerns each pattern being treated separately, $k=1, 2, \dots, N$). This objective function is minimized for each pattern separately. The necessary condition of minimum of V arises in the form

$$\frac{\partial V}{\partial u_{st}} = m u_{st}^{m-1} d_{st}^2 - \lambda \frac{\partial}{\partial u_{st}} \left(\bigvee_{i=1}^c u_{it} - 1 \right) = 0 \quad (10)$$

The second derivative involving the maximum operation requires special attention. We split the corresponding expression into two parts

$$\frac{\partial}{\partial u_{st}} \left(\bigvee_{i=1}^c u_{it} - 1 \right) = \frac{\partial}{\partial u_{st}} \left(\max \left(\bigvee_{i=1, i \neq s}^c u_{it}, u_{st} \right) \right)$$

Evidently, if u_{st} is bigger than the first term, the derivative returns 1. In the opposite case, the derivative is equal to zero as the term does not involve the variable over which the differentiation takes place. This binary nature of the derivative (that is equal to 0 or 1) may not be advantageous when running the optimization procedure. The primary concern is that when the derivative returns zero and this may occur over a broad range of the values of the variables (membership grades), then the iterative process could easily get 'stuck' and never reach a minimum of V (Pedrycz, 1991; Pedrycz, 2000; Pedrycz, Bargiela, 2001). To alleviate this shortcoming, we depart from the Boolean nature of the derivative and replace it by a multivalued (fuzzy) condition of inclusion describing a degree to which $\bigvee_{i=1, i \neq s}^c u_{it}$ is included in u_{st} .

The inclusion condition is quantitatively captured by means of the multivalued implication operation (\rightarrow)

$$\psi_{st} = \left(\bigvee_{i=1, i \neq s}^c u_{it} \right) \rightarrow u_{st}$$

in which we define

$$a \rightarrow b = \sup \{c \in [0,1] \mid a \otimes c \leq b\}, \quad a, b \in [0,1]$$

with " \otimes " being a t-norm (Butnariu, 1993; Pedrycz, 1991). In virtue of the multivalued implication defined above, two cases hold. If u_{st} exceeds $\bigvee_{i=1, i \neq s}^c u_{it}$ then the value of the implication is equal to 1 (as could have been expected). Otherwise, the implication returns a truth value of inclusion of $\bigvee_{i=1, i \neq s}^c u_{it}$ in u_{st} . The lower the degree of this inclusion, the lower the value of the above expression. Computationally, we have to pick up a certain form of the implication operation; any multivalued implication

could be of interest. For instance, one may consider the Lukasiewicz implication defined as

$$a \rightarrow b = \begin{cases} 1 - a + b & \text{if } a > b \\ 1 & \text{if } a \leq b \end{cases}, \quad a, b \in [0, 1] \quad (11)$$

Using the above implication (11), we get the expression

$$\mu_{st}^{m-1} d_{st}^2 - \lambda \psi_{st} = 0 \quad (12)$$

and

$$u_{st} = \left(\frac{\lambda}{m} \right)^{1/(m-1)} \left(\frac{\psi_{st}}{d_{st}^2} \right)^{1/(m-1)} \quad (13)$$

Plugging (13) into the normalization condition,

$$\bigvee_{s=1}^c (u_{st}) = 1$$

we eliminate the Lagrange multiplier λ obtaining the final expression for the membership values

$$u_{st} = \frac{1}{\bigvee_{j=1}^c \left[\left(\frac{\psi_{jt}}{\psi_{st}} \right)^{1/(m-1)} \left(\frac{d_{st}}{d_{jt}} \right)^{2/(m-1)} \right]} \quad (14)$$

Regarding the second tasks of determining the prototypes of the clusters, the solution to this problem is straightforward and arises in the form

$$v_i = \frac{\sum_{k=1}^N u_{ik}^m d_{ik}^2}{\sum_{k=1}^N u_{ik}^m} \quad (15)$$

The complete clustering algorithm is summarized in Table 1.

Table 1. The logic-based clustering algorithm

Given: data set $\{x_1, x_2, \dots, x_N\}$ where $x_k \in \mathbf{R}^n$, $k=1, 2, \dots, N$

Assumed: initial partition matrix $U(0)$, fuzzification factor (m), distance function $\|\cdot\|$, termination condition ($\epsilon > 0$)

repeat

 compute prototypes

$$v_i = \frac{\sum_{k=1}^N u_{ik}^m d_{ik}^2}{\sum_{k=1}^N u_{ik}^m}$$

$i=1, 2, \dots, c$, and partition matrix

$$u_{st} = \frac{1}{\bigvee_{j=1}^c \left[\left(\frac{\psi_{jt}}{\psi_{st}} \right)^{1/(m-1)} \left(\frac{d_{st}}{d_{jt}} \right)^{2/(m-1)} \right]}$$

$s=1, 2, \dots, c$, $t=1, 2, \dots, N$

until a termination condition ($\epsilon > 0$) has been satisfied

Returning to the general logic decomposition of data (4), it is worth stressing that we are concerned with the family of the partition matrices

$$\mathbf{U} = \{U \mid 0 < \sum_{k=1}^N u_{ik} < N \text{ for all } i = 1, 2, \dots, c \text{ and } \sum_{i=1}^c (u_{ik}) = 1 \text{ for all } i = 1, 2, \dots, N\}$$

(in fact, one may even emphasize the role of a particular s -norm by using the explicit notation $\mathbf{U}(s)$).

This general version gives rise to the optimization problem (here we have already converted it into the version involving the Lagrange multiplier)

$$V = \sum_{i=1}^c \sum_{k=1}^N u_{ik}^m d_{ik}^2 - \lambda \left(\sum_{i=1}^c u_{ik} - 1 \right) \quad (16)$$

The necessary condition of the minimum of V expresses in the form

$$\frac{\partial V}{\partial u_{st}} = m u_{st}^{m-1} d_{st}^2 - \lambda \frac{\partial}{\partial u_{st}} \left(\sum_{i=1}^c u_{it} - 1 \right) = 0 \quad (17)$$

The detailed formulas depend upon the form of the s-norm used above that need to be specified in advance. For instance, if we adopt the probabilistic sum ($asb = a+b-ab$, $a, b \in [0,1]$), we get

$$\frac{\partial V}{\partial u_{st}} = m u_{st}^{m-1} d_{st}^2 - \lambda \frac{\partial}{\partial u_{st}} (A_{st} + u_{st} - A_{st} u_{st} - 1)$$

where

$$A_{st} = \sum_{\substack{i=1 \\ i \neq s}}^c u_{it}$$

that leads to the expression

$$\frac{\partial V}{\partial u_{st}} = m u_{st}^{m-1} d_{st}^2 - \lambda (1 - A_{st})$$

(one should note that the computational complexity involved here is higher than in the max version discussed before).

The possibilistic clustering as introduced in (Krishnapurnam, 1995) addresses an important issue of relaxing the hyperplane constraint (expressing that the sum of membership grades is equal to 1) and admitting that

$$\max_i u_{ik} > 0$$

for any $k=1,2,\dots,N$. Note however that this formulation does not lead to *any* logic – based interpretation of the resulting clusters. In this sense, any comprehensive comparison of the possibilistic clustering and the approach presented here is not pertinent.

The logic-based clustering can be extended by associating it with an idea of concept formation (description), cf. (Pedrycz, 1998). It directly augments the original formulation discussed so far. The starting point is a fuzzy set - concept articulated in terms of some membership function B defined in some output space Y of finite dimensionality, $B = [b_k]$, $k=1, 2, \dots, N$. For instance, if the output space describes a price of real estate, such concept would represent an “affordable real estate”. This

concept can be described as a fuzzy set defined by some membership function, say a linearly decreasing function with high membership grades attached to low prices. Now we are interested in the formation of such descriptor in the form of some clusters - information granules that are defined in the multidimensional input space. This generalizes the previous problem to the form

$$\bigcup_{i=1}^c u_i = B \quad \text{that is} \quad \bigcup_{i=1}^c u_{ik} = b_k \quad (18)$$

where b_k is a membership grade of B for the k -th pattern. The semantics of (4) is governed by the logic expression

$$b_k = u_1 \text{ or } u_2 \text{ or } \dots \text{ or } u_c \quad (19)$$

stating that the concept (B) is a sum of the constructed fuzzy sets. To accommodate this extension, the previous clustering algorithm is modified as to the computations of the membership functions (partition matrix). The formula for the partition matrix reads in the form

$$u_{st} = \frac{b_t}{\bigvee_{j=1}^c \left[\left(\frac{\psi_{jt}}{\psi_{st}} \right)^{1/(m-1)} \left(\frac{d_{st}}{d_{jt}} \right)^{2/(m-1)} \right]} \quad (20)$$

The calculations of the prototypes of the clusters are the same as expressed in (15).

In all experimental studies, we use the maximum function as a model of the s-norm. To illustrate the idea of the logic-based clustering, we consider 24 one-dimensional data points:

(1.0, 0.8, 0.9, 1.1, 7.5, 8.1, 8.4, 9.9, 7.8, 8.0, 3.9, 4.5, 4.6, 3.8, 2.9, 5.0, 4.9, 5.2, 5.0, 4.5, 2.1, 2.2, 2.0, 2.0).

The clustering is carried out for $m = 4$ and the data set is partitioned into two clusters ($c=2$). The standard Euclidean distance is used to express the distance between the data. Figure 1 shows the membership grades of the patterns in the two clusters. Noticeably, the data close to the border exhibit membership values in the other cluster elevated to the level in the range from 0.3 to 0.4.

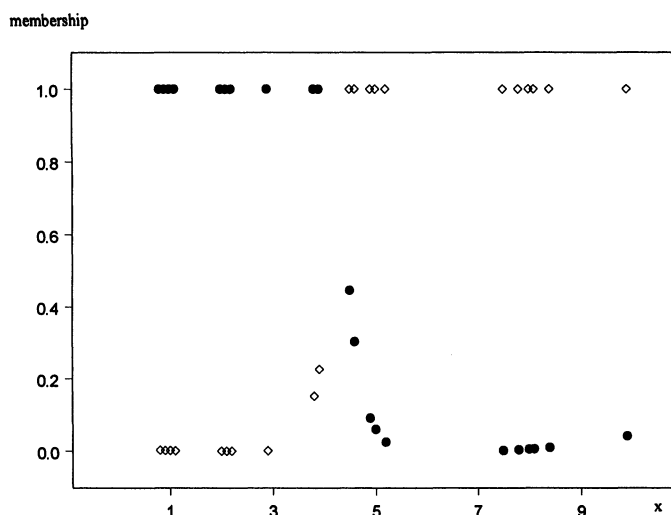


Figure 1. Membership grades of the clusters produced by the logic-based fuzzy clustering.

It is instructive to contrast the logic-based partition with the results obtained with the use of the standard FCM, refer to Figure 1, 2 and 3.

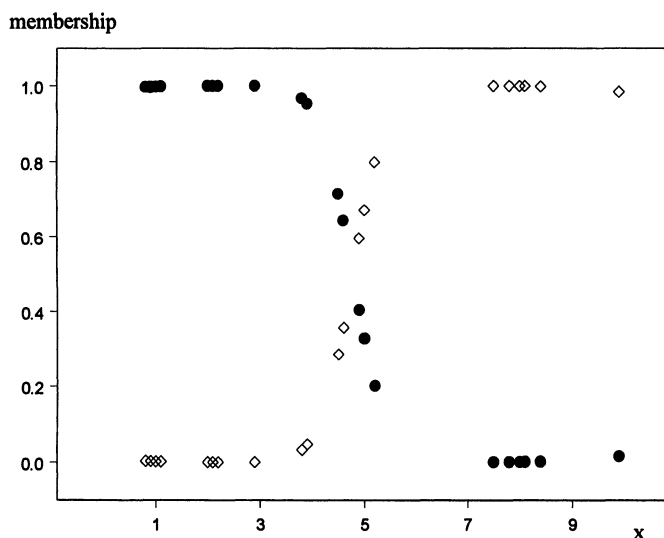


Figure 2. Membership grades of the patterns generated by the FCM method.

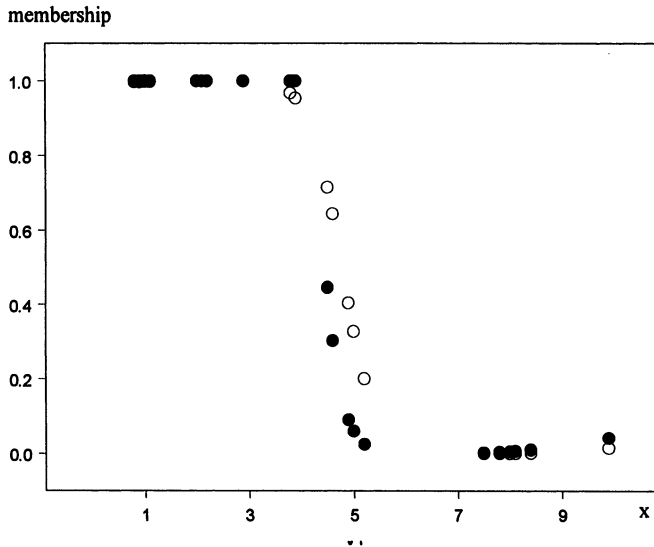


Figure 3. Contrasting membership grades of the logic-based clustering and the FCM method
solid dots: logic - based clustering, empty dots: FCM algorithm.

The difference between the logic-oriented clustering and the FCM is significant. FCM clustering identifies elements with partial membership to each cluster. The logic-based fuzzy clustering clearly allocates a pattern to a single cluster with the maximal grade. In this sense, the group assignment calls for a different interpretation, as the allocation of membership is very much different. The logic-based clustering assigns the pattern to the class without any doubt (as it produces a membership grade equal to 1) but at the same time issues an "overlap" signal that raises warning about some other membership assignment to this pattern to the remaining clusters. The FCM approach is oriented towards membership *sharing* and in this way it is more difficult to make any decision as to group allocation. In particular, it is not obvious (at least immediately) how to judge if the pattern should be regarded as a sort of bridge between two or more clusters. Furthermore to complete such group assessment, one needs a certain measure of overlap between clusters (Backer, 1995; Bezdek, 1981) (that may not be unique, anyway) to identify potential outliers.

9.3 EXPERIMENTAL STUDIES

In this section, we experiment with one of the well-known data sets available on the WWW from the Machine Learning Group at the University of California at Irvine (<http://www.ics.uci.edu/~mllearn/>) that is a Boston housing data. More specifically,

we consider 250 first patterns existing in the dataset. The number of the clusters is set to 2 and the distance is a weighted (normalized) Euclidean distance. The fuzzification factor "m" is equal to 4. The logic-based fuzzy clustering gives rise to the prototypes v_1 and v_2 , Figure 4:

$v_1 = [0.867600 \ 1.105489 \ 14.134078 \ 0.163700 \ 0.593915 \ 6.066246 \ 87.127876 \ 2.901850 \ 4.662981 \ 354.526154 \ 18.267658 \ 360.775665 \ 15.473553 \ 20.652462]$

$v_2 = [0.207897 \ 17.103449 \ 6.392184 \ 0.058829 \ 0.470909 \ 6.527604 \ 53.214638 \ 4.898777 \ 4.554705 \ 296.694977 \ 17.625763 \ 387.471558 \ 8.773052 \ 27.167801]$

To visualize the results, we confine ourselves to the last two variables, that is a lower status of the population (given in %) (x_1) and a price of real estate (x_2). The prototypes shown in Figure 4 reflect the structure of the data set (Figure 5), by being associated with regions of the highest density of patterns.

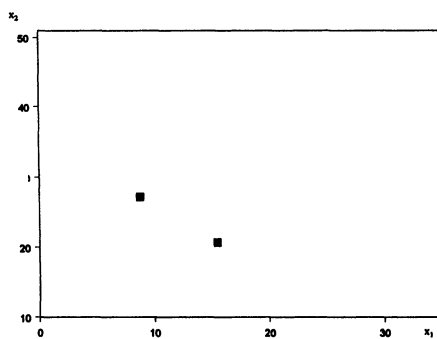


Figure 4. Prototypes of the logic-based prototypes in a two-dimensional space.

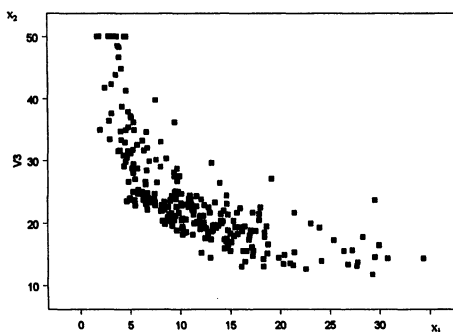


Figure 5. The Boston housing data set in the coordinates of percentage of lower status population and price of real estate.

The two clusters are visualized in Figure 6 (darker points correspond to higher values of membership grades). Noticeably, as we are confined only to two variables, there is some extra scattering of the patterns with high membership values in the clusters that are located not close to the centers of the clusters.

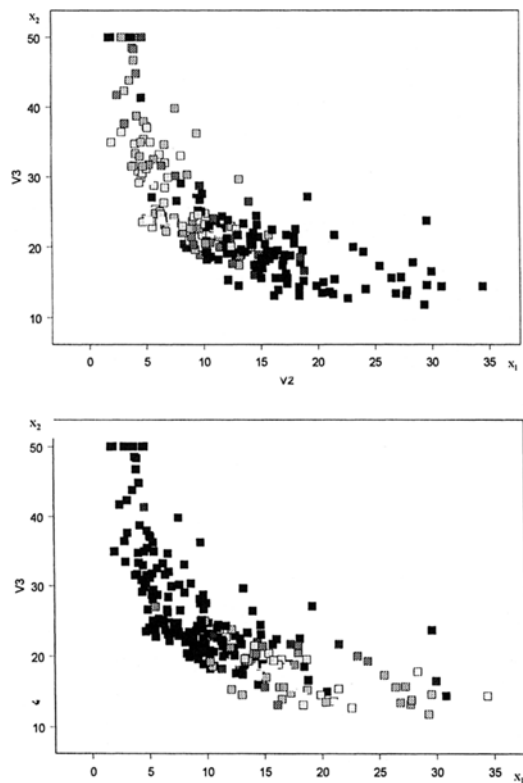


Figure 6. Visualization of the clusters generated by the logic-based clustering.

Figure 7 illustrates the level of interaction between the clusters. These are computed by summing up membership grades in each column of the partition matrix and subtracting one. The darker the color, the higher the interaction realized by the pattern.

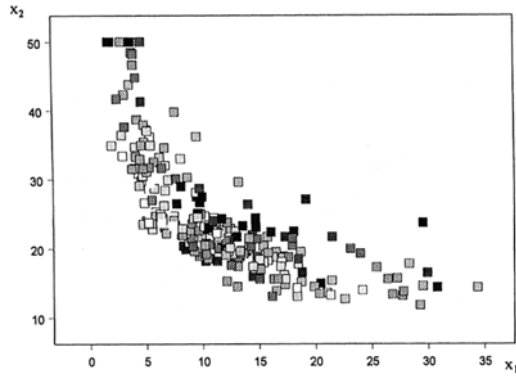


Figure 7. Interaction between the clusters.

As expected, most of the interaction occurs for the patterns located in-between the prototypes of the clusters. To carry out a comparative analysis, the same data set was processed by the FCM clustering (the method uses the same values of the parameters of the algorithm as set up in the previous experiments). Now the prototypes come in the form

$v_1 = [0.956589 \ 0.707430 \ 15.233346 \ 0.109893 \ 0.612766 \ 5.996574 \ 90.077660 \ 2.678182 \ 4.597565 \ 368.363831 \ 18.392817 \ 358.687531 \ 16.105392 \ 19.387083]$

$v_2 = [0.157886 \ 19.339594 \ 5.826880 \ 0.029371 \ 0.461537 \ 6.535730 \ 48.598164 \ 5.104898 \ 4.367391 \ 291.459045 \ 17.589252 \ 389.100769 \ 8.186273 \ 27.251997]$

and are broadly similar to the two representatives determined in the logic-based clustering. The two resulting clusters are visualized in Figure 8.

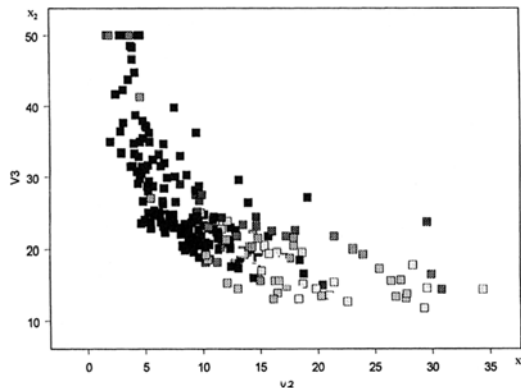


Figure 8(a). Clusters constructed with the aid of the FCM algorithm (darker color identifies higher membership values).

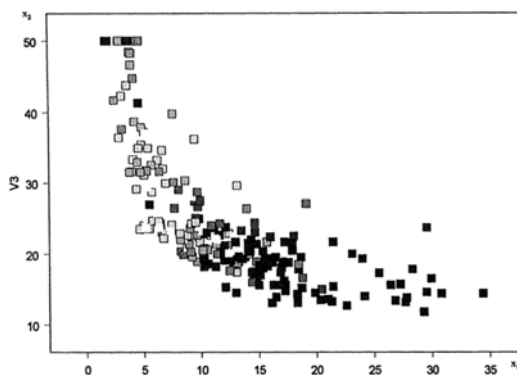


Figure 8(b). Clusters constructed with the aid of the FCM algorithm (darker color identifies higher membership values).

While the above visualization enables appreciation of the general cluster topologies the comparative analysis of the results is better accomplished in the space of membership grades of the logic-based and FCM partition matrices. Figure 9 relates these membership values computed for the same data point. Interestingly, when the membership values obtained in the FCM method exceed 0.5, the membership values in the logic-based clusters are essentially equal to 1. On the other hand when the membership values fall below 0.5, we get more dispersion between the corresponding membership values obtained via the two clustering methods. In other words, the link between the two clustering results is weak in quantitative terms. Nevertheless a general monotonicity trend holds: higher values of the membership grades in one method coincide with the higher membership grades in the other.

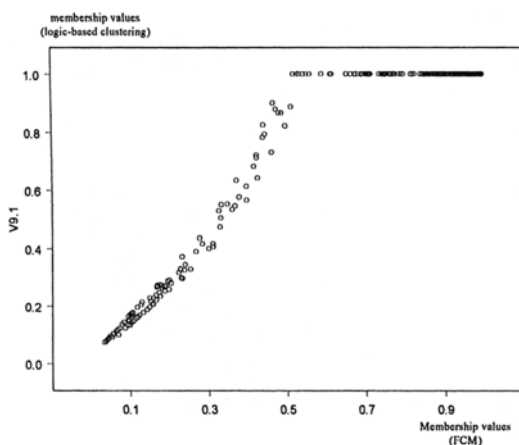


Figure 9. Membership values derived in two clustering methods under discussion.

Another interesting option is to look into the relationships between the two clusters obtained by the same clustering method. Figure 10 pinpoints such relationships for the logic-based clustering. It is noticeable that the membership grades could change independently from each other (as the maximum operation is quite *permissive* with this regard). Note, however, that there is a significant density of patterns in the regions where the membership grade to one cluster is equal 1 and the membership to the second one is quite low. The logic nature of the constraint on the partition matrix provides us with flexibility not encountered in the FCM approach where given one membership grade, the other one is lined up by the identity constraint.

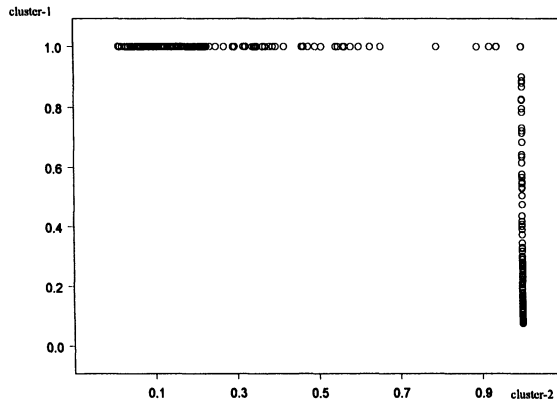


Figure 10. Distribution of membership grades of patterns in two clusters generated by the logic-based clustering.

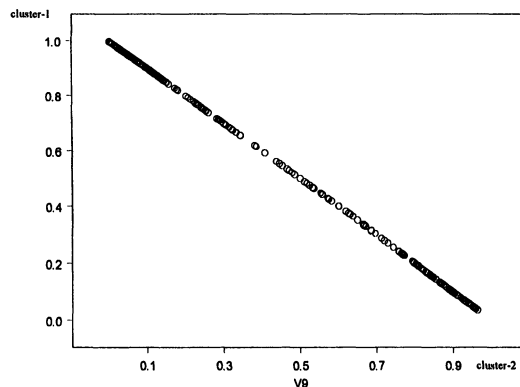


Figure 11. Distribution of membership grades of patterns in two clusters generated by the FCM clustering.

By contrast, the FCM algorithm distributes the membership values along a single line (as could have been anticipated), see Figure 11. Given membership values in one cluster, the second one determined in a unique way as its complement. A high density of points occurs at the two ends of the line and this is a direct consequence of the existence of the two well-separated clusters.

9.4 CONCLUSIONS

The logic-based fuzzy clustering discussed in this chapter allows processing of constraints on the partition matrix regarded as a union of fuzzy sets generated through fuzzy clustering. The logic nature of the clustering makes it profoundly distinct from other forms of the FCM. The importance of this generalization is twofold

- We can treat clusters formed in this way as fuzzy relations to which regular logic-based operations encountered in fuzzy sets fully apply.
- The logic constraints imposed on the membership functions can be immediately exploited in the construction of higher-level concepts; in this sense we may view the clusters as primitive entities contributing to building such concepts.

This perspective allows us to gain a better insight into the logic fabric of the granular structure of data. The detailed algorithm has been provided for the selected s-norm used as the model of the logic aggregation of the clusters. Variants of the algorithm can be derived using alternative s-norms.

REFERENCES

- Backer, E. (1995), *Computer-Assisted reasoning in Cluster Analysis*, Prentice Hall, N. York.
- Bargiela, A., Pedrycz, W., Hirota, K. (2002), Logic-based granular prototyping, *Soft Computing and Intelligent Systems Conference, SCIS 2002*, Tsukuba, Japan, Oct. 2002.
- Bargiela, A., Pedrycz, W., Hirota, K. (2002), Data granulation through optimization of similarity measure, *Archives of Control Sciences*, to appear.
- Bezdek, J.C. (1981), *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, N. York.
- Butnariu, D., Klement, E.P. (1993), *Triangular Norm – Based Measures and Games with Fuzzy Coalitions*, Kluwer Academic Publishers, Dordrecht.
- Dave, R.N., Krishnapuram, R. (1997), Robust clustering methods: a unified view, *IEEE Trans. on Fuzzy Systems*, vol. 5, 2, 2, 270-293.

Klir, G.J., Yuan, B. (1995), *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hall, Saddle River, NJ.

Krishnapuram, R., Keller, J.M. (1995), A possibilistic approach to clustering, *IEEE Trans. on Fuzzy Systems*, vol. 1, 98-110.

Pedrycz, W. (1991), Processing in relational structures: fuzzy relational equations, *Fuzzy Sets and Systems*, 40, 77-106.

Pedrycz, W. (1998), Conditional fuzzy clustering in the design of radial basis function neural networks, *IEEE Trans. on Neural Networks*, vol. 9, 601- 612

Pedrycz, W. (2000), Fuzzy relational equations: bridging theory, methodology, and practice, *Int. J. of General Systems*, 29, no.4, 529-554.

Pedrycz, W., Bargiela, A. (2001), Information granulation: A search for data structures, *Knowledge-based Engineering Systems KES 2001*, Osaka, October 2001, 1147-1151.

Zimmermann, H.J. (1991), *Fuzzy Set Theory and Its Applications*, 2nd edition, Kluwer Academic Publishers, Boston, MA.

SEMANTICAL STABILITY OF INFORMATION GRANULES

The essence of information granules is their embodiment of abstract perceptions of real-world phenomena. Information granules are useful if they are comprehensible, reflect the experimental evidence and have a stable interpretation as entities that bridge experimental reality with the subjective and ultimately observer-based judgement about the environment. Once being semantically stable, information granules could be viewed as architecture-independent. The proposed algorithmic environment supporting this concept dwells on the ideas of statistical inference that helps quantify stability thorough a nonparametric testing. The χ^2 goodness-of-fit test is used here as a validation mechanism. First, the study elaborates on the formation of information granules and concentrates on the descriptive and prescriptive ways of their design. In the sequel, it is revealed how these two ways interact with the construction of stable information granules. A number of experimental studies are also included.

10. 1 INTRODUCTION

Information granulation is a process of abstraction in which we identify elements that are similar in terms of their functional, spatial or temporal proximity. In the previous chapters we have discussed granulation algorithms in two broad frameworks: that of set (interval) analysis and fuzzy sets / fuzzy logic. When granulating information in the setting of fuzzy sets, we are concerned with the development of membership functions or their counterparts, namely characteristic functions (Pedrycz, 1998; Zimmermann, 1991). Interestingly, since the very inception of the technology of fuzzy sets, there have been a number of methodological and ensuing algorithmic points of view dealing with the development of membership functions (Dubois, 1983; 1994; Medasani, 1995; Saaty, 1975). Several main points of view at fuzzy sets can be identified. They are radically different as far as the underlying interpretations of fuzzy sets and the development techniques are concerned (Dubois, 1983; 1994). They involve likelihood view, random set view and the typicality view. The development of

information granules can be completed along the line of so-called prescriptive design and its complementary view ensuing their descriptive construction. The prescriptive design relies heavily on the perception of meaningful granules being expressed by a certain observer (agent). The descriptive design dwells on the utilization of data (usually of numeric character) that are embraced to form information granules. One can refer here to granular models, especially fuzzy models constructed with the aid of clustering techniques, see e.g., (Abe, 1995; Pedrycz, 1998; Bargiela, Pedrycz, 2001, 2002; Takagi, 1985). The prescriptive development of information granules lead to the danger of constructing a fictitious notion that has nothing to do with real world or is only loosely related to the essence of the concept. As such, these information granules are somewhat marginal as far as their future usage (and eventual calibration or refinements) is concerned. Consider, for instance, that one of the fuzzy sets used in the fuzzy model has never been “activated” by any experimental evidence (e.g., numerical data). If this is the case, the part of the model exploiting such a fuzzy set will neither be constructed (estimated, optimized, etc.), nor validated once the fuzzy model has been completed. In this sense, the fuzzy set is a fictitious entity not supported by any experimental numerical evidence. On the other hand, the approach that dwells exclusively on experimental data (descriptive approach) can also be affected in a negative way. Essentially, the development of the membership function could be “blind” as far as the meaning of the information granule is concerned. Subsequently, the construct is dangerously close to the purely statistical investigation of numeric data not being provided with any guidance as to the semantics of the fuzzy set to be constructed.

Interestingly, an evident trend encountered in many constructs, especially neurofuzzy systems (Pedrycz, 1998; Pedrycz, Bargiela, 2002) is that of the optimization method induced specificity of the information granules. It is noticeable that the membership functions (being developed through the learning process) are the result of the overall learning in the neurofuzzy system. Even a slight change in the neural part of the architecture (say, the number of nodes in the network, number of hidden layers or alike) will affect the obtained information granules (e.g., the membership function). This makes the information granules very architecture specific - the feature we are usually not very comfortable with. The essence of information granulation calls for a degree of stability of information granules. We usually require that the granules are *stable* in the sense of holding their meaning across a broad range of experimental evidence (say, numeric data). While the design of the information granules has to take into account the experimental evidence, they should not be driven by any architectural environments and therefore constitute some generic entities (building blocks) that could be easily exploited in a vast array of granular systems (including rule-based systems, classifiers, controllers, etc.).

The above observations imply the main focus of this chapter namely a formal definition of semantical stability of information granules and the algorithmic quantification of this concept.

10. 2 INFORMATION GRANULATION: DESIGN AND VALIDATION

The key phases of the overall process of information granulation are portrayed in Figure 1. It consists of the following components:

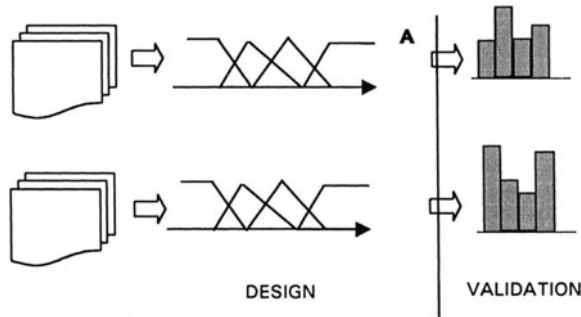


Figure 1. Information granulation: design and validation phases.

- ❖ **Design** The environment of experimental evidence \mathcal{E} is split into the training and testing part. For each of them the family of information granules is constructed. Denote them by $\mathbf{A} = \{A_1, A_2, \dots, A_n\}$ and $\mathbf{B} = \{B_1, B_2, \dots, B_n\}$, respectively. While the granules in \mathbf{A} and \mathbf{B} may look quite different (e.g., we may encounter different membership function with different values of their parameters), we assume that the number of the granules is the same, $\text{card}(\mathbf{A}) = \text{card}(\mathbf{B})$.
- ❖ **Validation** The two families of the information granules designed above are sought to be equivalent (and therefore the granulation accomplished in this way regarded stable) if they are equivalent in terms of the available experimental evidence. As the gathered evidence is usually numeric, the equivalence is expressed through the use of some statistical testing (nonparametric testing). This way of validation (statistical inference) is pertinent to processing numeric data occurring in \mathcal{E} .

The above depicts a general methodology and embraces a spectrum of algorithmic approaches as far as the design of the information granules goes as well as the ensuing validation procedures are concerned.

Bearing in mind the above scheme, we can introduce the following definition of stability of information granulation

Definition Information granules arising from experimental evidence E are said to be semantically stable if the families of information granules A and B are statistically equivalent (in terms of some statistical test involving the elements of E).

Note that, in essence, the notion of stability is perceived in the setting of information granules and does not exhibit any relationship with the notion of stability in the sense encountered in control systems (viz. stability of closed loop control systems).

As a follow-up of this definition, we may say that if A and B are statistically equivalent then this pertains to their stability. In other words, these two families of information granules belong to the same class G . Obviously, G may include a series of other families (information granules) beside A and B . This definition of stability is related to the given experimental evidence.

The general scheme outlined above concerns one-dimensional information granules. More specifically, the validation phase is possible because we can easily establish a one-to-one correspondence between the elements of A and B . Unfortunately, this is not the case in the multidimensional case where such correspondence cannot be developed and the validation phase being realized in a straightforward manner. We then refine the concept and propose the following development scheme whose main functional modules are depicted in Figure 2.

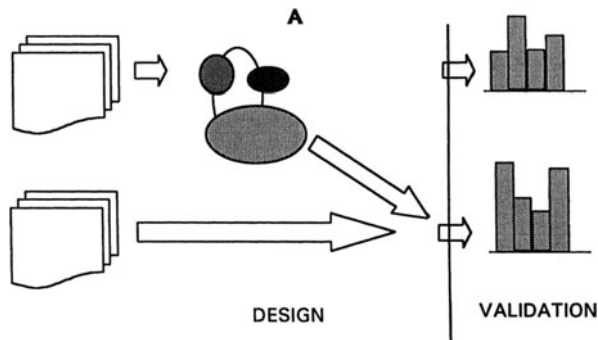


Figure 2. The development of information granules in the multidimensional case; note the use of clustering techniques in information granulation.

As before, the experimental framework of evidence E is divided into the training and testing set. The main difference in comparison to the one-dimensional case is that now we develop a single collection of information granules as opposed to two collections utilised in the previous scenario. In the sequel, A is validated with the aid of both training and testing sets and such validation is completed in the statistical setting.

The two environments of information granulation being of interest in this study involve fuzzy sets and sets (more specifically, intervals). Furthermore, we consider sets to be derived from some information granules represented by fuzzy sets. In this sense, we are interested in inducing sets from fuzzy sets. The approximation technique addressing this transformation is discussed in the ensuing section.

10.3 SET APPROXIMATION OF FUZZY SETS

Fuzzy sets are generalizations of sets. Therefore a set-based approximation of fuzzy sets makes sense. Here we express the approximation of fuzzy sets by sets as a certain optimization problem. The fundamental finding can be formulated as follows

Proposition

Consider a unimodal normal fuzzy set A defined in \mathbf{R} with a continuous membership function. Its best approximation (in the sense of the Minkowski distance) arises as set A^* with the characteristic function that is an $\frac{1}{2}$ -cut of A

$$A^*(x) = A_{1/2}(x)$$

Proof. Let us consider the performance index Q expressing a distance between A and A^* ,

$$Q(\alpha) = \int_a^b |A(z) - A^*(z)|^p dz \quad (1)$$

The exponent 'p' with $p > 1$ standing in the performance index gives rise to the Minkowski distance between the fuzzy set and its approximation. Bearing in mind the unimodality of the fuzzy set (which is quite general, anyway), we rewrite (1) in the form of the series of integrals

$$Q(\alpha) = \int_a^{x_0} A^p(z) dz + \int_{x_0}^m (1 - A(z))^p dz + \int_m^{y_0} (1 - A(z))^p dz + \int_{y_0}^b A^p(z) dz \quad (2)$$

Note that the optimal threshold level (α) identifies two elements in the universe of discourse X , say x_0 and y_0 , as seen in Figure 3 and already used in the above formula. That is, $A(x_0) = \alpha$, $A(y_0) = \alpha$.

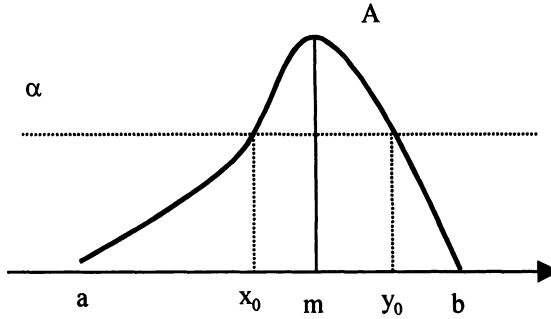


Figure 3. Approximating fuzzy set (A) by A_α through its α -cut optimization.

The optimization of Q carried out with respect to α is equivalent to the optimization of Q with respect to x_0 and y_0 meaning that

$$\text{Min } Q(\alpha) = \text{Min } Q(x_0, y_0)$$

The necessary conditions leading to the minimum of Q read as

$$\frac{\partial Q}{\partial x_0} = pA^{p-1}(x_0) - p(1 - A(x_0))^{p-1} = 0 \quad (3)$$

and

$$\frac{\partial Q}{\partial y_0} = p(1 - A(y_0))^{p-1} - pA(y_0)^{p-1} = 0 \quad (4)$$

Solving (3) we get $A(x_0) - 1 + A(x_0) = 0$. This leads to $A(x_0) = \frac{1}{2}$ viz. x_0 is a point where the membership function attains $\alpha = \frac{1}{2}$. Analogously, we handle (4) which again leads to the same result as before, namely $A(y_0) = \frac{1}{2}$.

Interestingly, the threshold (that is $\frac{1}{2}$) does not depend on the form of the membership function. The generality of the finding (that is quite intuitive) clearly points out how fuzzy sets can be converted into sets. An important observation is that not all fuzzy sets are equally easy to approximate: the higher the performance index $Q(\frac{1}{2})$, the more *difficult* is to approximate the fuzzy set under discussion and more questionable this approximation is. In general, it is advisable to consider the value of the performance index along with the resulting set approximation of the fuzzy set under discussion. As an example, let us discuss how a triangular fuzzy set is approximated, see Figure 4.

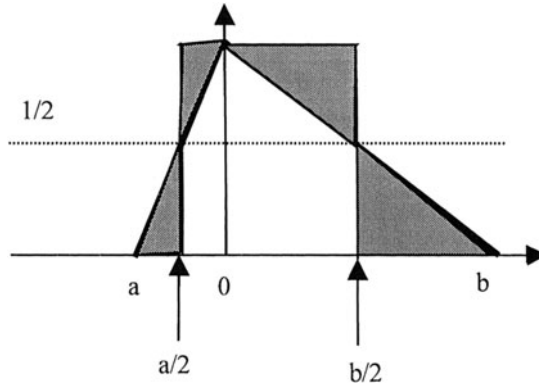


Figure 4. Triangular fuzzy set and its set-based approximation. The shadowed regions quantify the approximation error.

Put also $p = 1$ (so we are concerned with the Hamming distance). The approximation error is visualized in the form of the four triangular regions whose total area amounts to

$$Q = \frac{1}{4} (a+b)$$

This error is a linear function of the lower and upper bound of the fuzzy set. We can rewrite Q in the form underlining its relation with the support of A

$$Q = \frac{1}{4} \text{supp}(A) + \frac{1}{2} a$$

Where $\text{supp}(A) = b-a$.

10. 4 ALGORITHMIC ISSUES OF INFORMATION GRANULATION: DESIGN AND VALIDATION

In this section, we expand on the algorithmic fabric of the general scheme discussed in Section 10.2. First, we concentrate on the design of information granules elaborating on one-dimensional and multivariable constructs. Next, a statistical approach to the analysis of stability of information granules is presented in detail.

The design of fuzzy sets - information granules

As underlined, the development of fuzzy sets is completed in many different ways. In general, these could be based exclusively on perceptions of observers or dwelled upon numeric data (descriptive approach). In the design of such information granules, we are interested in the construction of individual granules as well as

maintaining integrity of the entire collection of granules (**A**, **B**, etc.). The collections of fuzzy sets (**A** = { A_1, A_2, \dots, A_n }) being fuzzy partitions are of particular interest in this setting. Let us recall that these are fuzzy sets whose membership values sum up to 1 for any element of the universe of discourse **X** over which A_i 's are defined, namely

$$\sum_{i=1}^c A_i(x) = 1 \text{ for } x \in X \quad (5)$$

In particular, triangular membership functions with an 1/2 overlap existing between any two adjacent fuzzy sets constitute a fuzzy partition. This particular level of overlap translates into the following equality

$$\sum_{k=1}^N \sum_{i=1}^c A_i(x_k) = N \quad (6)$$

Where x_k is an element of the experimental evidence **E** and "N" denotes a number of all data in **E**. If the fuzzy sets exhibit an underlap (in the sense their sum is less than one), we get

$$\sum_{k=1}^N \sum_{i=1}^c A_i(x_k) < N \quad (7)$$

Moreover, for the overlap (for the sum exceeding 1), we obtain

$$\sum_{k=1}^N \sum_{i=1}^c A_i(x_k) > N \quad (8)$$

The fundamental property of the 1/2 overlap shows up a nice relationship with the distribution of sets approximating fuzzy sets. The two cases (overlap equal to 1/2 and under 1/2) are illustrated in Figure 5.

The choice of the fuzzy sets (in this case triangular membership functions) could be exclusively driven by perception of a human observer. It is the designer of a system, developer of the model who decides to utilize a collection of information granules as the entities being of interest in problem solving. In such a sense, the experimental evidence (and numeric data) come into play in an indirect fashion, see (Dubois, 1983; Saaty, 1975). It could well be that the data have a very limited impact on the final shape of the membership functions.

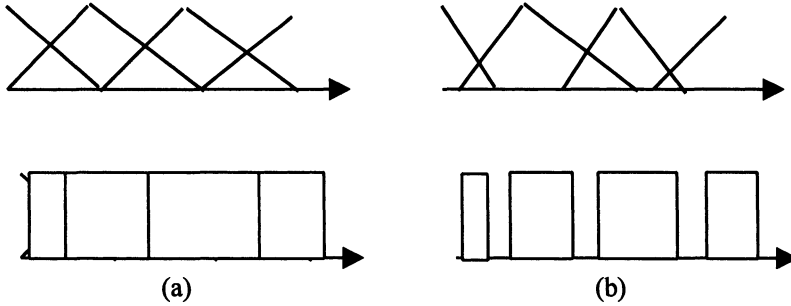


Figure 5. Triangular fuzzy sets and the resulting sets forming their approximations; note their distribution resulting due to different levels of overlap: 1/2 overlap (a), and an overlap less than 1/2 giving rise to gaps between the sets (b).

On the other hand, one can easily encounter situations in which the developments of the membership functions are driven exclusively by experimental data. Fuzzy clustering (Bezdek, 1981) is an excellent example of the approach falling under this category. Let us remind that clustering techniques are exclusively descriptive methods. Membership functions are constructed in a form of a partition matrix. The FCM algorithm being commonly used is an objective function-based approach to data analysis. The objective function Q reads as

$$Q = \sum_{i=1}^c \sum_{k=1}^N u_{ik}^m \| \mathbf{x}_k - \mathbf{v}_i \|^2 \quad (9)$$

where " c " denotes the number of clusters, $\| \cdot \|$ is a distance function, \mathbf{v}_i s are the prototypes of the clusters and $U = [u_{ik}]$, $i = 1, 2, \dots, c$, $k = 1, 2, \dots, N$ is a partition matrix containing all membership grades. The fuzzification factor (m) is used to "control" the shape of the membership functions; the value m equal to 2.0 is in common use. The clusters are constructed through the minimization of Q . In this sense we anticipate that the clusters reflecting the structure of the data should minimize a dispersion of data around the prototypes treated as focal points of the entire data set. In contrast to the prescriptive approach to the construction of the information granules, fuzzy clustering helps deal with multivariable information granules (being more specific, fuzzy relations) and those are data - driven. Interestingly, the fuzzy relations form a fuzzy partition and thus satisfy (6). The form of the membership function that can be easily reconstructed with the use of the formula

$$u_j(\mathbf{x}) = \frac{1}{\sum_{i=1}^c \left(\frac{\|\mathbf{x} - \mathbf{v}_j\|}{\|\mathbf{x} - \mathbf{v}_i\|} \right)^{2/m-1}} \quad (10)$$

$j=1, 2, \dots, c$ and depends upon the distribution of the prototypes (centroids) $\{\mathbf{v}_j\}$ and is to some extent predetermined by some other parameters of the algorithm (such as fuzzification factor, m , and the form of the distance function $\|\cdot\|$). The derivation of (10) is a consequence of the following optimization task

$$\text{Min } Q = \sum_{i=1}^c u_i^m \|\mathbf{x} - \mathbf{v}_i\|^2$$

realized with respect to $u_1(\mathbf{x}), u_2(\mathbf{x}), \dots, u_c(\mathbf{x})$ and subject to the membership grade constraint

$$\sum_{i=1}^c u_i(\mathbf{x}) = 1$$

where \mathbf{x} is a data point of interest; note that the entire optimization is centered around a collection of the given prototypes (that have been constructed through the previous phase of data clustering).

The validation phase

Are the collections of information granules **A** and **B** stable? Intuitively speaking, stability of information granules relates with their similarity. In particular, one may think of exploiting a well defined conceptually and algorithmically apparatus of statistics. The validation phase is cast in the framework of nonparametric statistics, especially the one of χ^2 goodness-of-fit, see (Sheskin, 1997). The formulation is produced as follows. Recall that **A** is formed in some way with the use of the training portion of the experimental evidence, say **E**. Similarly, the information granules in **B** are developed in the setting of the remaining portion of **E** (that is a testing portion). It is worth stressing that by proceeding with the validation phase we have not confined ourselves to any specific way of building the information granules. It could well be that the construction of **A** and **B** may dwell on some different formalism. Now we look at **E** through the already constructed information granules. Denote by $n_i(\mathbf{A})$ and $n_i(\mathbf{B})$ the following sums (that expresses a cumulative level of "activation" of the corresponding granules either in **A** or **B** by the numeric data)

$$n_i(\mathbf{A}) = \sum A_i(x_k)$$

More precisely, $N(\mathbf{A}) = \{n_i(\mathbf{A})\}$ forms a histogram of \mathbf{E} developed in the framework of \mathbf{A} . The same interpretation holds for $N(\mathbf{B}) = \{n_i(\mathbf{B})\}$. Proceeding with the line of such statistical inference, the following null hypothesis is formed

$$H_0: A_i = B_i \text{ for all information granules in } \mathbf{A} \text{ and } \mathbf{B} \quad (11)$$

The test of the hypothesis (which, in essence, pertains to the stability of the information granules), we use the χ^2 goodness-of-fit. The value of the pertinent statistic is computed as follows

$$\chi^2 = \sum_{i=1}^c \frac{(n_i(\mathbf{B}) - n_i(\mathbf{A}))^2}{n_i(\mathbf{A})} \quad (12)$$

To reject the null hypothesis the value of the statistic (11) must be equal to or greater than the critical value χ_{crit}^2 . This critical value is usually taken at p equal to 0.05 or 0.01. The number of degrees of freedom is equal to $c-1$.

10.5 EXPERIMENTS

This section summarizes the results of experiments completed for synthetic and real-world data. In the latter case, we use the well-known Boston housing data available on the WWW (Blake). In all experiments, we split the data into two subsets of equal size. The first one is a training set while the second one is treated as the testing one. Both one dimensional case and multidimensional cases are considered.

Synthetic one dimensional data

The data set consists of numeric data drawn from one-dimensional normal distribution. The data set, Figure 6, shows a partition into two equal subsets where \mathbf{A} and \mathbf{B} are formed, respectively. For given fuzzy sets, their set-based approximation uses their 1/2 cuts (as discussed in Section 3).

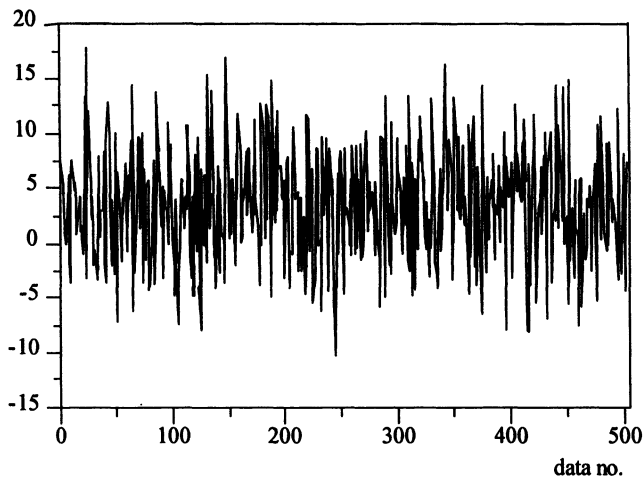


Figure 6. One-dimensional synthetic data set.

The values of the χ^2 goodness-of-fit statistic are summarized in Table 1 and 2. They concern both fuzzy set and set-based information granulation.

Number of information granules	Goodness of fit χ^2
3	0.8371
4	3.3233
5	8.0221
6	10.7830
7	11.1312
8	12.7340
9	15.2725
10	17.9966

Table 1. Goodness-of-fit χ^2 statistic for information granulation carried out in terms of fuzzy sets for selected number of the information granules (c). Fuzzy sets are described by triangular membership functions with 1/2 overlap and are distributed uniformly over X.

The values of the statistic χ^2 are compared with the critical values of this test and subsequently summarized in Figure 7. Two selected critical values being in common use (0.05 and 0.01) are exploited. The results reveal a profound difference: fuzzy set-based granulation contributes to a far higher stability than the one observed for

set-based granulation. The goodness of fit for these granules remains far below the critical values for the entire range of the cluster numbers. This contrast with sets where only a three-and five -element collection of granules does not exceed the critical value for $p=0.01$.

Number of information granules	Goodness of fit χ^2
3	2.1471
4	15.9692
5	12.6993
6	22.7983
7	27.6222
8	33.1006
9	27.8704
10	31.3194

Table 2. Goodness-of-fit χ^2 for information granulation carried out in terms of sets for selected number of the information granules (c).

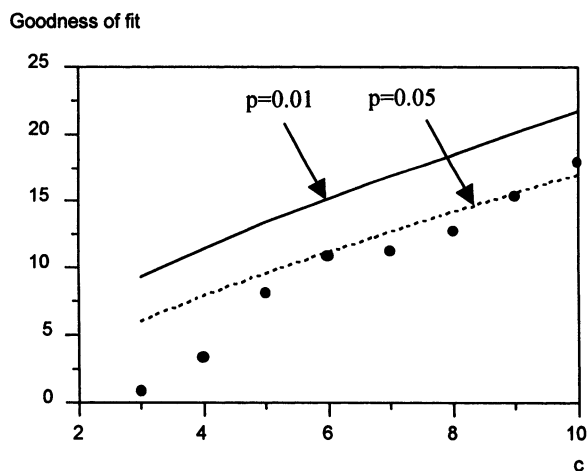


Figure 7(a). The values of the χ^2 goodness-of-fit static and the critical values at p equal to 0.05 and 0.01 for a fuzzy set-based granulation

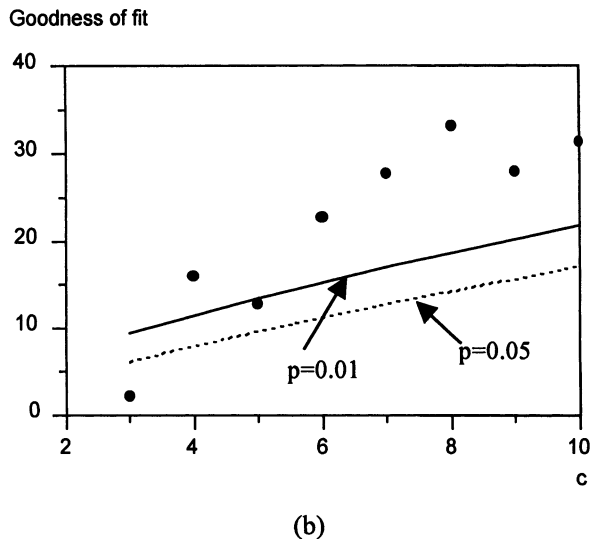


Figure 7(b). The values of the χ^2 goodness-of-fit static and the critical values at p equal to 0.05 and 0.01 for a set-based granulation.

Real-world data

The real-world data concern a characterization of Boston real estate market where each data is described by 14 features including price, crime rate, distance from main working centers, student-teacher ratio, etc. One-dimensional and multidimensional cases are considered.

One-dimensional case

We consider one of the attributes of the data, namely crime rate per capita and granulate it using triangular fuzzy sets with an overlap of $1/2$ between adjacent granules. The triangular fuzzy sets are distributed uniformly across the universe of discourse. Subsequently, these fuzzy sets are approximated by sets.

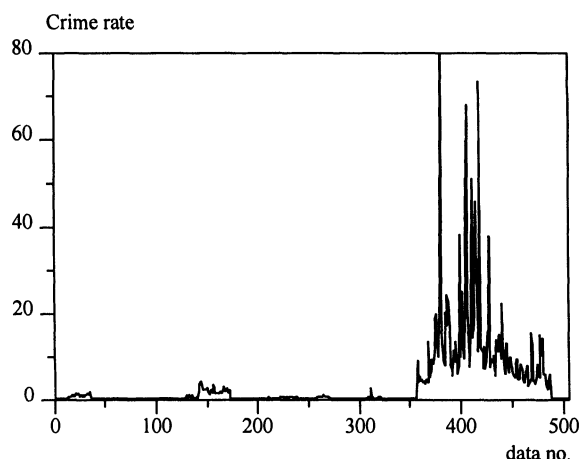


Figure 8. Experimental data set of crime rate per capita.

The values of the χ^2 statistic are summarized in Table 3. Moreover, they are illustrated in Figure 9 showing these values vis-à-vis the critical values at $p = 0.05$ and 0.01 . Similarly, the same experiment is carried out for the set-based granulation. The results are given in Table 4 and Figure 10, respectively (the critical value is again p set to 0.05 and 0.01).

Number of triangular fuzzy sets	Goodness of fit χ^2
3	7.08
4	10.47
5	13.49
6	16.78
7	19.17
8	18.01

Table 3. Goodness of fit χ^2 for fuzzy granules.

Number of information granules (sets)	Goodness of fit χ^2
3	19.82
4	12.91
5	11.11
6	28.24
7	40.67
8	53.98

Table 4. The χ^2 values for set-based information granules.

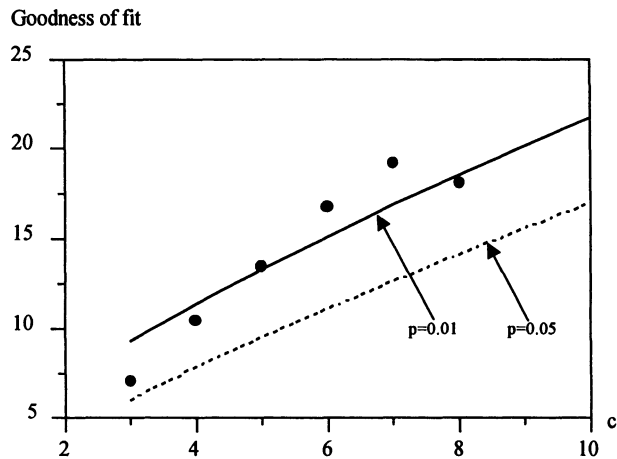


Figure 9. The values of the goodness-of-fit statistic versus critical values for fuzzy set based granulation.

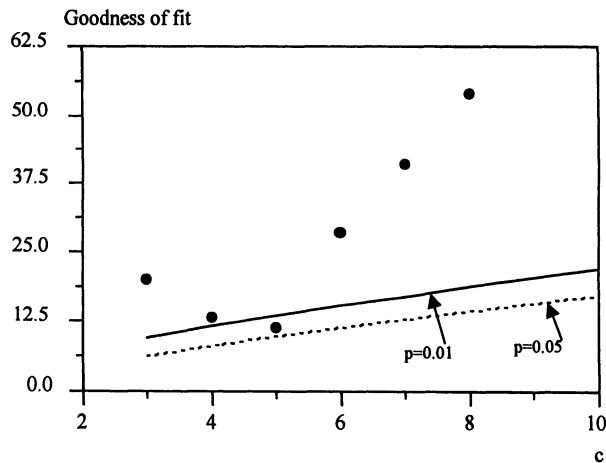


Figure 10. The values of the goodness-of-fit statistic versus critical values for set-based granulation.

These results provide a useful insight into the nature of granulation. Fuzzy sets are again more stable than sets. As a matter of fact, the latter are characterized by the values of χ^2 far above the critical values. For $p = 0.01$ the stability is retained for $c = 3$ and 4 information granules.

After experimenting with other variables, a similar pattern of behavior could be observed. In some cases, though, we encounter a significant instability of the information granules with high values of the χ^2 statistic. This could have been expected as in this design of the information granules we have not confined in any particular way to the experimental data standing behind the summarization (description) provided by information granules.

The multidimensional case

In this experiment, we exploit all 14 features of the patterns. The training data are split into information granules using the FCM method in the form introduced in Section 5. The number of information granules varies from $c = 2$ up to $c = 9$. The fuzzification factor (m) was kept equal to 2 across all the experiments.

For illustrative purposes, the prototypes for $c = 7$ and $c = 3$ are summarized in Table 5. Primarily, one can observe that the prototypes are not evenly distributed across the respective universes of discourse.

prototype 1								
0.5380	5.3601	10.9548	0.0792	0.5415	6.1748	77.6408	3.5866	4.6129
339.6086	18.7864	376.0323	13.6021	21.5829				
prototype 2								
0.2835	15.7301	7.3998	0.0585	0.4850	6.4467	55.0994	4.7671	4.4448
301.9492	17.7517	383.7372	9.5480	25.9747				
prototype 3								
0.5364	5.4104	10.9272	0.0794	0.5411	6.1764	77.4924	3.5951	4.6136
339.2417	18.7758	376.0675	13.5736	21.6160				
prototype 4								
0.2736	16.9264	7.2188	0.0563	0.4818	6.4656	53.7751	4.8582	4.4417
301.5663	17.7108	383.9885	9.3320	26.1945				
prototype 5								
1.4711	2.3458	17.4917	0.1070	0.6710	6.0032	90.4666	2.3977	4.7580
383.5582	16.4848	340.0468	15.1061	20.7352				
prototype 6								
0.3250	12.4790	8.0694	0.0670	0.4963	6.3827	59.7987	4.4851	4.5024
305.9082	17.8865	382.5281	10.3414	25.1649				
prototype 7								
0.2454	21.4130	6.6638	0.0495	0.4722	6.5246	49.5329	5.1623	4.4601
302.0559	17.5625	384.6492	8.6774	26.8543				

prototype 1								
0.4651	9.0457	9.5197	0.0822	0.5218	6.2932	67.8496	4.0520	4.6018
319.3425	18.0253	377.5480	11.6871	23.8738				
prototype 2								
0.2485	20.7916	6.6331	0.0445	0.4721	6.5153	48.8586	5.1584	4.3855
299.5522	17.5691	384.6142	8.5864	26.7972				
prototype 3								
0.7163	5.1287	12.6430	0.0961	0.5695	6.1570	80.3332	3.3103	4.6294
349.7631	18.2903	368.2460	14.0265	21.6180				

Table 5. Prototypes of the clusters for $c = 7$ and $c = 3$.

Then the same information granules are investigated in the setting of the testing data. The counting bins are filled out using the σ -count of the data falling within the scope of the corresponding information granules (fuzzy relations). The values of the χ^2 statistic are computed and compared with their critical values. The results are summarized in Table 6.

Number of information granules	Goodness of fit χ^2
2	1.5462
3	1.1993
4	2.0877
5	2.1442
6	2.1039
7	2.0501
8	1.7217
9	1.3944

Table 6. The values of χ^2 for selected values of "c".

As becomes obvious from Table 6, for $c = 2$ up to 9, information granules remain semantically stable (the values of the statistics being substantially lower than the critical ones).

Now instead of linguistic information granules, we consider the set-based mechanism of granulation meaning that the fuzzy partition is replaced by its induced two-valued counterpart (in this case, the allocation to a specific granule is completed based on the maximal value of the membership grade, say $i_0 = \arg \max u_i(x)$ where x denotes a pattern under discussion. Then the relevance of such information granules is evaluated by computing the values of the statistics. The obtained results reveal that in all cases (viz. for the varying number of information granules), the

computed values of the statistics exceeds substantially the critical values and the hypothesis about the stability of such information granules has to be rejected.

The obtained values of the statistic for the relation-based and fuzzy relation-based information granules are easy to contrast. The continuous boundaries of fuzzy sets and an overlap occurring between them contribute to high extent to the stability of the granules. This, in turn, becomes an evident testimony to the superiority of fuzzy sets over sets when it comes to concept description. This particular advantage is very apparent in the setting discussed here.

10. 6 CONCLUSIONS

This chapter discussed an important concept of semantical stability of information granules. The algorithmic environment supporting this concept dwells on the ideas of statistical inference and helps quantify stability thorough nonparametric testing. We showed that the χ^2 goodness-of-fit test can be used as a tool for determination of semantical stability of granules. While, in essence, the findings in this chapter are highly intuitive, we moved on beyond that by providing an algorithmic tool for quantitative assessment of stability. An important result is a finding that fuzzy sets are far more stable than sets (used for their approximation). This is, obviously, highly appealing and speaks for itself when we anticipate further usage of fuzzy sets in any detailed constructs such as rule-based systems, classifiers, or control systems. One should stress that the notion of information granularity is a fundamental concept and the level of granulation itself is pivotal to any further pursuit. In particular, one is interested in forming granules that are stable so that they could be viewed as architecture-independent concept. While being tuned-up to reflect the efficacy of the experimental data, such stable information granules promote design of effective granular systems.

REFERENCES

- Abe, S., Lan, M.S. (1995), Fuzzy rules extraction directly from numerical data for function approximation, *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 25, 1, 119-129.
- Bargiela, A., Pedrycz, W. (2001), Granular clustering with partial supervision, *European Simulation Multiconference ESM2001*, Prague, June 2001, 113-120.
- Bargiela, A., Pedrycz, W. (2002), A model of granular data: A design problem with the Tschebyshev-based clustering, *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2002*, Hawai, 578-583.
- Blake, C., Keogh, E., Merz, C.J., *UCI repository of machine learning databases* [<http://www.ics.uci.edu/~mllearn/MLRepository.html>], Irvine, CA: University of California, Department of Information and Computer Science.

Bezdek, J.C. (1981), *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, N. York.

Dubois, D., Prade, H. (1983), Unfair coins and necessity measures: towards a possibilistic interpretation of histograms, *Fuzzy Sets and Systems*, 10, 15-20.

Dubois, D., Prade, H. (1994), Fuzzy sets - a convenient fiction for modeling vagueness and possibility, *IEEE Trans. on Fuzzy Systems*, 2, 16-21.

Medasani, S., Kim, J., Krishnapuram, R. (1995), Estimation of membership functions for pattern recognition and computer vision, In: *Fuzzy Logic and Its Applications to Engineering, Information Sciences, and Intelligent Systems*, K.C. Min and Z. Bien (eds.), Kluwer Academic Publishers, 45-54.

Pedrycz, W., Gomide, F. (1998), *An Introduction to Fuzzy Sets. Analysis and Design*. MIT Press, Cambridge, MA.

Pedrycz, W. (1998), Conditional fuzzy clustering in the design of radial basis function neural networks, *IEEE Trans. on Neural Networks*, vol. 9, 601- 612.

Pedrycz, W., Vukovich, G. (1999), Data-based design of fuzzy sets, *Journal of Fuzzy Logic and Intelligent Systems*, Vol. 9, No. 3.

Pedrycz, W., Smith M.H., Bargiela, A. (2000), Granular clustering: A granular signature of data, *Proc. 19th Int. (IEEE) Conf. NAFIPS'2000*, Atlanta, July 2000, 69-73.

Pedrycz, W. (2001), Fuzzy equalization in the construction of fuzzy sets, *Fuzzy Sets and Systems*, 119(2), 321-327.

Pedrycz, W., Bargiela, A. (2002), Granular Clustering: A granular signature of data, *IEEE Trans. on Systems Man and Cybernetics*, Vol 32, No. 2, 212-224.

Saaty, T.L. (1975), *The Analytic Hierarchy Process*, Mc Graw-Hill, N. York.

Sheskin, D., (1997), *Handbook of Parametric and Nonparametric Statistical Procedures*, CRC Press, Boca Raton, Fl.

Takagi, T., Sugeno, M. (1985), Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 15, 1, 116-132.

Zimmermann, H.J. (1991), *Fuzzy Set Theory and Its Applications*, 2nd edition, Kluwer Academic Publishers, Boston, MA.

PART III

GRANULAR WORLD COMMUNICATIONS



COMMUNICATIONS BETWEEN GRANULAR WORLDS: FUNDAMENTALS

11. 1 INTRODUCTION

Granular Worlds (GWs) are computational and conceptual entities grounded in the language of information granules. Let us recall that GWs are defined as the following structures

$$GW = \langle X, \mathcal{G}, \mathbf{A}, \mathbf{C} \rangle \quad (1)$$

with \mathbf{C} denoting a family of communication procedures that help the worlds exchange information granules and make them “understood” and be interpreted in a given world. Schematically, as shown in Figure 1, we can portray these procedures as an interface between the core of the processing going on in the given world and the surrounding environment.

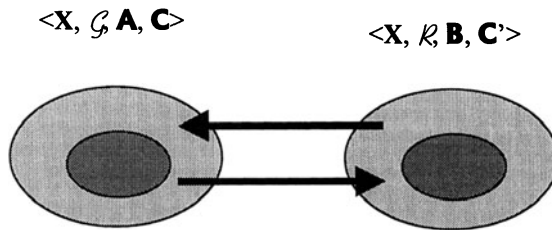


Figure 1. Communication between granular worlds: shown are their communication layers (\mathbf{C} and \mathbf{C}').

The granules exhibit different levels of granularity. The formal framework of a specific granular computing could vary from environment to environment that is GW to GW. Granular worlds have to establish communication namely exchange

information granules, interpret them in the language specific to the given granular world, do necessary processing and finally release the result (that is again a certain information granule) to the external world in the format that becomes accepted there. For instance, a granular world of information granules realized via fuzzy sets “talks” to the world of probabilistic granules. This means that the basic communication scheme has to assure that the worlds can understand each other and produce meaningful pieces of information. Several essential categories of the communication problems are worth visualizing:

- Communication between two worlds $\langle X, \mathcal{G}, \mathbf{A} \rangle$ and $\langle X, \mathcal{G}, \mathbf{A}' \rangle$ developed in the same conceptual framework (say sets, fuzzy sets, rough sets, etc.) that exhibit a significant difference in the level of granularity where the cardinality of elements in \mathbf{A} is very much different from those in \mathbf{A}'
- Communication between the worlds $\langle X, \mathcal{G}, \mathbf{A} \rangle$ and $\langle X, \mathcal{K}, \mathbf{B} \rangle$ where the formal frameworks \mathcal{G} and \mathcal{K} are different (e.g., the worlds of fuzzy sets and rough sets).
- Communication of any granular world with the world of numeric entities (that can be treated here as a granular world exhibiting the highest possible level of granularity; in essence it is a limit case for all granular worlds, no matter what formalism they dwell on).

As a prerequisite, we start with a generic scenario where a granular world exploits fuzzy sets. Some other granular worlds use sets. We are interested in building some communication mechanisms between them. This study helps us explore a variety of possible communication mechanisms that are similar in other circumstances.

11.2 REPRESENTATION OF FUZZY SETS IN THE SET-THEORETIC FRAMEWORK

The problem of communication as already discussed is directly related with the issue of representation of any fuzzy set by a collection of sets. More formally, we pose the problem as follows: represent a given fuzzy set R by a finite and fixed family of sets $\mathbf{A} = \{A_i\}$, $i=1, 2, \dots, c$. In essence, this representation problem splits into two tasks: (i) an approximation of R to be completed in terms of these sets and (ii) its subsequent reconstruction during which R becomes “recovered”. Alternatively, one can also view these two phases as a set-based compression and its decompression. Figure 2 visualizes a multimodal fuzzy set with a superimposed family of sets (intervals).

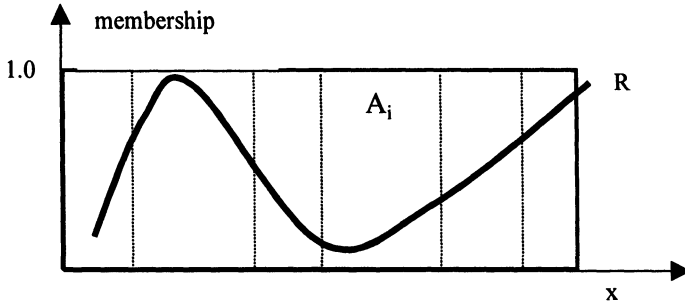


Figure 2. Fuzzy set R with a collection of superimposed sets A_i .

The first step in the compression of R is to represent it via a collection of A_i s. While there are a number of viable options, a sound representation scheme is the one based on the use of possibility measures. First, the possibility measure is logically sound and results in interesting mechanisms of matching between fuzzy sets (fuzzy relations) involved in the compression model. Second, an inverse problem occurring in this setting can be formulated by means of fuzzy relational equations (Di Nola et al., 1989) that help formulate and solve such problems. We compute a possibility measure of A_i with respect to R. As A_i is a set, the computations are straightforward. Let us start with the general formula

$$\lambda_i = \sup_{x \in X} [A_i(x) \text{t} R(x)] \quad (2)$$

The above expression expresses an extent to which A_i and R overlap (coincide logically). In case of sets, this expression returns a maximal membership grade of R that occurs within the support of A_i , namely $\sup_{x \in A_i} R(x)$. As a matter of fact, as A_i is a set, the choice of the t-norm in (2) does not matter; a minimum operation would be the simplest choice.

The transformation of this nature returns a collection of λ_i s that form a basis for the reconstruction of R. More specifically, we have a collection of “c” requirements

-Given $\{A_i\}$ and the associated sequence of possibility values $\{\lambda_i\}$, “reconstruct” (determine) R

There is no unique solution to this problem. There is however a maximal solution (Di Nola et al., 1989) whose construction is supported by the theory of fuzzy relational equations (as a matter of fact, (2) is a sup-t composition of R and A_i). The membership function of this maximal fuzzy set \hat{R}_i induced by the reads as

$$\hat{R}_i(x) = A_i(x) \rightarrow \lambda_i = \begin{cases} 1 & \text{if } A_i(x) \leq \lambda_i \\ \lambda_i & \text{otherwise} \end{cases} \quad (3)$$

Referring to Figure 3, we note that \hat{R}_i contains the original fuzzy set (where the containment is sought in the sense of fuzzy set inclusion).

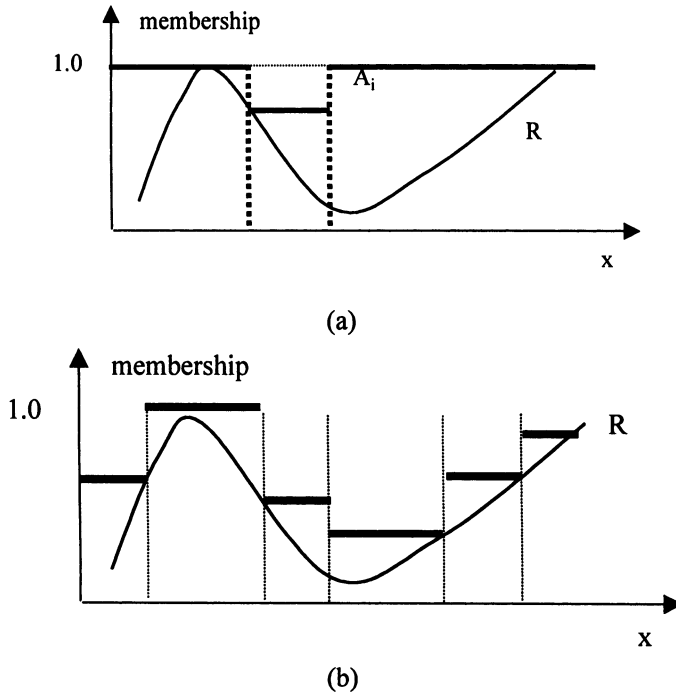


Figure 3. Reconstruction of R (solid staircase line) with the use of a single set (A_i) (a) and family $\{A_i\}$ (b).

When using the entire family of A_i (that leads to the intersection of \hat{R}_i s) we obtain

$$\hat{R} = \bigcap_{i=1}^c \hat{R}_i \quad (4)$$

(refer also to Figure 3). Interestingly, the reconstructed fuzzy set exhibits a stairwise membership function where the height of the individual jumps and their distribution across the space depends on the distribution of A_i s. From the theoretical point of view that arise in the realm of fuzzy relational equations, we note that we are dealing

here with a system of equations $\lambda_i = \text{Poss}(R, A_i)$, $i=1,2,\dots, c$ to be solved with respect to R for λ_i and A_i provided.

Similarly, we can determine the necessity measure of the A_i with respect to R following the well-known expression (Dubois and Prade, 1980)

$$\mu_i = \inf_{x \in X} [(1 - A_i(x))sR(x)] \quad (5)$$

The reconstruction of R on the basis of the necessity measure is again supported by the theory of fuzzy relational equations (Di Nola et al, 1989); here the minimal solution to (5) reads in the form

$$\tilde{R}_i(x) = (1 - A_i(x)) \varepsilon \mu_i = \begin{cases} \mu_i, & \text{if } 1 - A_i(x) < \mu_i \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

The result of the compression and decompression mechanism realized by the necessity measure is shown in Figure 4.

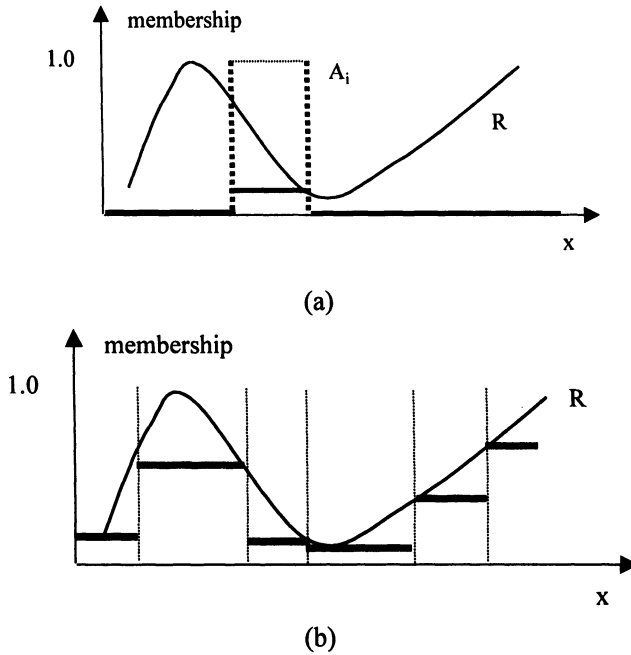


Figure 4. The mechanism of reconstruction realized in terms of the necessity measure : single set A_i (a) and a family of sets (b).

To complete reconstruction of R , we take a union of the minimal solutions that leads to the expression

$$\tilde{R} = \bigcup_{i=1}^c \tilde{R}_i \quad (7)$$

The result is again a stairwise type of the membership function; the distribution of the steps and their heights is implied by the collection of the sets being used in the granulation. Referring to the theory of fuzzy relational equations we note that we have a collection of constraints $\mu_i = \text{Nec}(A_i, R)$ to be solved with respect to R . As \hat{R} and \tilde{R} form the upper and lower bound of the original fuzzy set (see Figure 5), we can conclude that the decompression realized for a fuzzy set returns an interval fuzzy set, $\langle R_-, R_+ \rangle$. Interestingly, the communication between the set-based environment results in a granular construct that is no longer an element of the same granular environment – we conclude that the communication process does not operate in a closed environment.

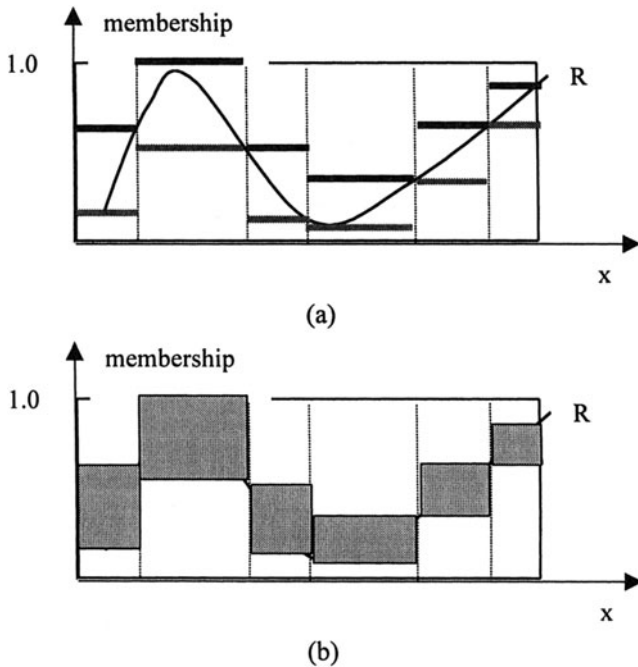


Figure 5. Reconstruction of R : upper and lower bound of the reconstructed information granule (a) leading to an interval-valued fuzzy set (b).

The above model of reconstruction can be easily expanded to cases where instead of sets $\{A_i\}$ we deal with a collection of fuzzy sets; the same formulas apply. The visible qualitative difference will be in the form of the bounds of the reconstructed fuzzy set R whose lower and upper bound get smoother. An interesting problem arises as to optimization of the bounds that is how to make them as tight as possible. One can either consider a certain class of fuzzy sets (R) and discuss how to optimize (say, distribute) A_i s that come from some family of membership functions. The other task would be to study a class of fuzzy sets R whose reconstruction can be realized with the upper and lower bound that are tight enough. Both of these categories of problems are inherently linked with the granularity of the family of fuzzy sets or more directly the number of sets or fuzzy sets in \mathbf{A} .

From a functional point of view, we may also look at the representing X via \mathbf{A} and its further reconstruction as a process of transmitting X through a communication channel where the representation of X corresponds to the coding of information while the decoding results in its reconstruction, see Figure 6.

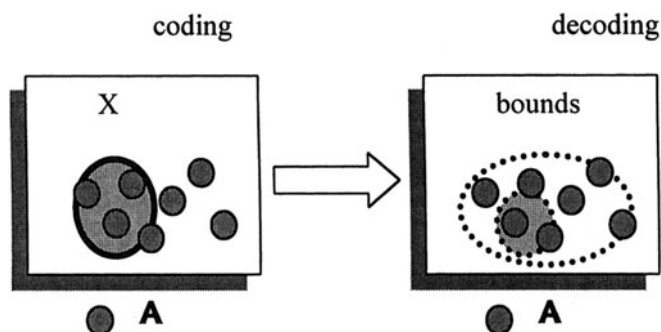


Figure 6. Coding and decoding mechanisms in the communication channel as an equivalent functional view at the reconstruction problem.

11.3 COMMUNICATION WITH A NUMERIC WORLD

Numeric world of analog signals (variables) is a limit case of all granular worlds, no matter what formalism we start with. While receiving an input datum X that is a number and processing it in the framework of some world $\mathbf{G} = \langle X, \mathcal{G}, \mathbf{A} \rangle$, we transform it into the format acceptable for internal processing of the world. This transformation is straightforward. The communication procedures that return the result of granular processing realized in \mathbf{G} are of interest, as here we have to produce a numeric value that is sought as a representative of an information granule.

Naturally, this transformation is not unique. At the same time, the mapping of this nature has a long tradition, especially in interval analysis and fuzzy sets.

Interval analysis and interval calculus are the crux of digital processing. Numeric data are converted to a digital (that is an interval format existing in this granular world) through a commonly used analog-to-digital conversion. Whatever computing takes place at the digital level, the results needs to be communicated to the external physical worlds (e.g., actuators existing in the process, etc.). Evidently, to transform an interval to a single number comes with an error. Simply, there is no unique element that fully represents an interval. The simplest choice one could make is to take a middle point of the interval x_0 , see Figure 7. The absolute difference $\varepsilon = |x - x_0|$ where $x \in [a, b]$, is viewed as an approximation error or, as usually referred to in digital signal processing (DSP), a quantization error.

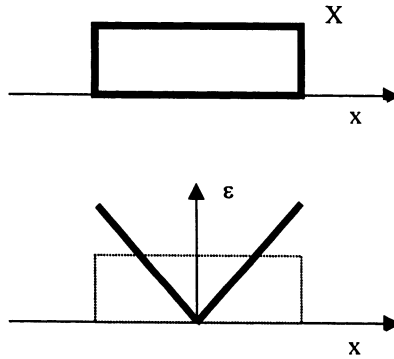


Figure 7. Selection of an optimal numeric representative of the interval and the resulting approximation error ε .

The magnitude of the quantization error depends on the size of the intervals or the number of bits used in the digital-to-analog converter.

Communication with the granular world of fuzzy sets $\langle X, \mathcal{G}, \mathbf{A} \rangle$, i.e., $\mathcal{G} = \mathcal{I}(X)$. In many applications of fuzzy sets (especially those in the realm of fuzzy control and fuzzy controllers) we communicate between the granular world of fuzzy sets by producing a single numeric value of control. Again there is no unique way of completing this transformation. We can refer to a number of communication procedures (**C**) encountered in the literature (Pedrycz and Gomide, 1998; Zimmermann, 2001). Noticeably, in the literature we often refer to them as defuzzification schemes, which are algorithms that remove “fuzziness” from the fuzzy set. Each of the communication procedures comes with some underlying

rationale and experimental evidence. A list of some of them along with auxiliary motivation is summarized in Table 1.

Communication procedure	Computations	Comments
Center of gravity	$\tilde{y} = \frac{\int_x A(y)ydy}{\int_x A(y)dy}$	Commonly used communication procedure that takes into account the shape of the membership function
Simplified center of gravity	$\tilde{y} = \frac{\sum_{i=1}^c w_i \bar{y}_i}{\sum_{i=1}^c w_i}$	It is a simplified version of the center of gravity which does not require integration
Maximal membership	$\tilde{y} = \arg \max_y A(y)$	The result is based on the maximal membership value and does not reflect the form of the membership function of B. Works well in case of a unimodal fuzzy set
Center of area (bisection)	$\int_{y:\tilde{y}<y} A(y)dy = \int_{y:\tilde{y}>y} A(y)dy$	The result bisects the area under the membership function

Table 1. A list of communication procedures (defuzzification schemes) between the world of fuzzy sets and numeric environment, the modal values of the fuzzy sets in **A** are denoted by $\bar{y}_1, \bar{y}_2, \dots, \bar{y}_c$. \tilde{A} is computed in the form where w_1, w_2, \dots, w_c are the levels of matching of A_i s and the numeric input, $w_i = A_i(x_0)$

As shown in this table, the communication usually involves a collection of fuzzy sets **A**. Their choice implies the performance of the communication procedure in the sense of the approximation error. In contrast to the interval-based granular world, we may end up with a zero approximation error (owing to the partial membership grades that allows us to discriminate between the elements of **X**). In particular, we can show that when dealing with **A** composed of triangular fuzzy sets with a $\frac{1}{2}$ overlap between two successive fuzzy sets, Figure 8, and using the center of gravity

method (see Table 1), we end up with the zero value of the approximation error (Pedrycz and Gomide, 1998).

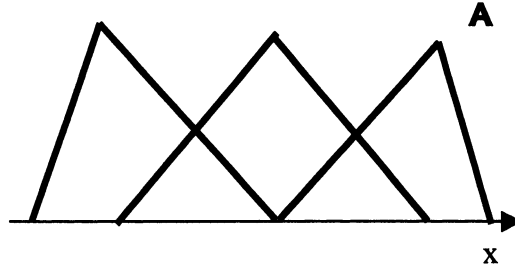


Figure 8. A collection of triangular fuzzy sets A producing a zero approximation error.

Considering the defuzzification expression in the form

$$\tilde{y} = \sum_{i=1}^c w_i \bar{y}_i$$

where w_i s sum up to 1, we observe that \tilde{y} is equal to the original numeric value. In other words, the communication becomes error-free. One should stress that if w_i are affected by error, the result will also come with some error. To quantify an impact w_i has on the result, let us compute the absolute value of the derivative $\frac{\partial \tilde{y}}{\partial w_i}$.

Bearing in mind that $y \in [\bar{y}_i, \bar{y}_{i+1}]$ we have the following relationship

$$\tilde{y} = w_i \bar{y}_i + (1 - w_i) \bar{y}_{i+1}$$

(where only two successive fuzzy sets are involved in the reconstruction process) and

$$\left| \frac{\partial \tilde{y}}{\partial w_i} \right| = | \bar{y}_i - \bar{y}_{i+1} |$$

meaning that the sensitivity depends on the differences between the modal values of the fuzzy sets. We note that this expression assumes constant values in-between the modal values of the fuzzy sets while the reduction of the sensitivity values can be realized by adding more fuzzy sets (as the differences between successive modal values get lower).

11. 4 CONCLUSIONS

Communication models between granular worlds become their essential feature. We have augmented the worlds by communication procedures (forming the communication layer **C**) and discussed several general problems to be dealt with. There are three categories of them: (a) communicating between worlds based on the same formalisms ($\mathcal{G} = \mathcal{B} = \mathcal{C}$ etc), (b) communication involving worlds where information granules originate from different frameworks, $\mathcal{G} \ \mathcal{B} \ \mathcal{C}$ etc. and (c) communication between granular worlds and the numeric world. The latter one is regarded as a limit case of all granular worlds so it becomes essential to develop a comprehensive picture of possible mechanisms of interaction. In this context, we clearly see that the size of information granules plays a pivotal role. In general, there is no hesitation when casting granules of high granularity in the framework of information granules of lower granularity (this situation occurs when a numeric datum is sent to any granular world; the understanding of it is straightforward). The opposite is not that obvious: when we communicate information granules to the numeric environment. As such environment requires granules of the highest granularity, the granules to be communicated by the other world need to be “upgraded” to the higher level of granularity. We showed that there is no unique way to realize that. The collection of defuzzification schemes is a testimony of developments in this direction. As a matter of fact, there is a general intuitively appealing conjecture of an irreversible degradation of granularity: in general, the granules whose granularity level has been reduced cannot be “recovered” (viz. transformed) to the granules of higher granularity.

REFERENCES

- Bortolan, G., Pedrycz, W.(1997), Reconstruction problem and information granularity, *IEEE Transactions on Fuzzy Systems*, 2, 234 - 248.
- Di Nola, A., S. Sessa, S., W. Pedrycz, W., E. Sanchez, E.(1989), *Fuzzy Relational Equations and Their Applications in Knowledge Engineering*, Kluwer Academic Press, Dordrecht.
- Dubois, D., Prade, H. (1980), *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, New York.
- Pedrycz, W., Gomide, F.(1998), *Fuzzy Sets*, MIT Press, Cambridge, MA.
- Valente de Oliveira, J. (1996), Sampling, fuzzy discretization, and signal reconstruction, *Fuzzy Sets and Systems*, 79, 151-161.
- Zimmermann, H.J. (2001), *Fuzzy Set Theory and Its Applications*, 4th Edition, Kluwer Academic Publishers, Boston, Dordrecht.

NETWORKING OF GRANULAR WORLDS: COLLABORATIVE CLUSTERING

12. 1 INTRODUCTION

In this chapter, we concentrate on the algorithmic aspects of networking of granular worlds. Each of the worlds focuses on its own computing agenda, say some tasks of intelligent data analysis and data mining. They operate on their own databases. Each of them concentrates on some specific processing activities that are usually cast in the framework of information granules. Here we assume that the formal framework of information granulation is the same across a network of granular worlds and are based on fuzzy sets. The objective of networking the granular worlds is to collaborate and exchange findings in order to support activities being in scope of each world. Intelligent agents and their collaboration over the Internet is an excellent testimony to such collaboration (Agah and Tanie, 1999; Genesareth and Ketchel, 1994; Loia and Sessa, 2001). In many areas of everyday activity various databases are constructed, used and maintained independent from each other. In each local environment, one tries to make sense of data by engaging in various activities of data mining and data analysis. The obtained results can be useful to such local community yet they could be of significant interest to the others. This triggers interest in a collaborative effort where the data mining activities could exploit several databases and the ensuing results benefit a larger circle of users. While it sounds appealing, one has to remember that sharing data, especially those of more confidential nature, is a genuine obstacle. This matter has to be taken seriously when moving along any collaborative pursuit in data analysis. Interestingly, granulation of information can be a viable option to make collaboration possible in presence of the confidentiality requirements.

This collaboration-driven task of data mining calls for an orchestrated effort and implies a highly collaborative nature of search for dependencies in data so that that such findings are common and relevant to all databases (as such discoveries of

global character are of genuine interest). To shed light on the spectrum of the processing problems, we identify possible scenarios along with existing drawbacks and envision potential mechanisms of collaboration

- Search for a common structure in databases Within a given organizational structure (company, network of sales offices, etc.), there are several local databases of customers (e.g., each supermarket generates its own database or a sales office maintains a local database of its customers). Generally, we can assume that all databases have the same attributes (features) while each database consists of different objects (patterns). To derive some global relationships that are common to all these databases, we should allow the databases to collaborate at the level of the patterns. Quite commonly, we may not be permitted to have access to all databases but eventually could be provided with some general aggregates (say, some synthetic indexes describing data; a mean value or median are a good example in this case). Refer to Figure 1 that illustrates the underlying concept. Bearing this in mind, we can talk about *vertical* (data based) collaboration in the process of knowledge elicitation (that is revealing a common structure in the data).

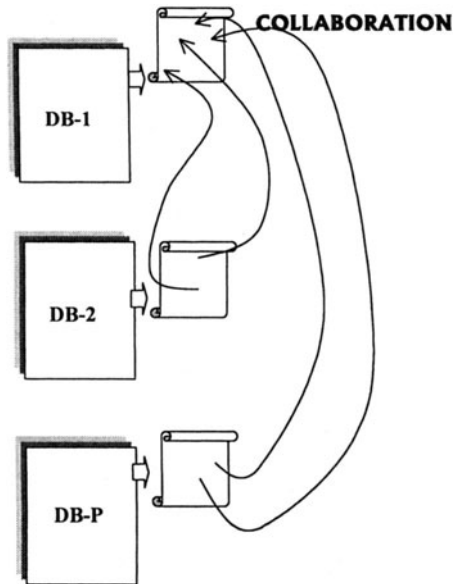


Figure 1. Vertical collaboration between databases at a local level; in each database objects are located in the same data space but deal with the different patterns.

- Security issues and discovery of data structures across different datasets. Consider now that information about the same group of clients is collected

in different databases where an individual company (bank, store, etc.) builds its own database. Because of confidentiality and security requirements, the companies cannot share information about clients in a direct manner. However all of them are vitally interested in deriving some associations that help them learn about clients (namely, identifying their profiles and needs). As they are concerned with the same population of clients, we may anticipate that the basic structure of the population of such patterns, in spite of possible minor differences, should hold across all databases. The approach taken in this case would be to build clusters in each database and exchange information at the level of the clusters treated here as information granules. Subsequently, we allow all collaboration processes to be realized at this particular level. In this manner, the security issues are not compromised while a sound mechanism of collaboration/interaction between the databases becomes established. Graphically, we can envision the situation of such collaboration as the one portrayed in Figure 2. Evidently, in this case we are concerned with a *horizontal* (that is feature-based or attribute - oriented) collaboration completed in the search for the overall data structure.

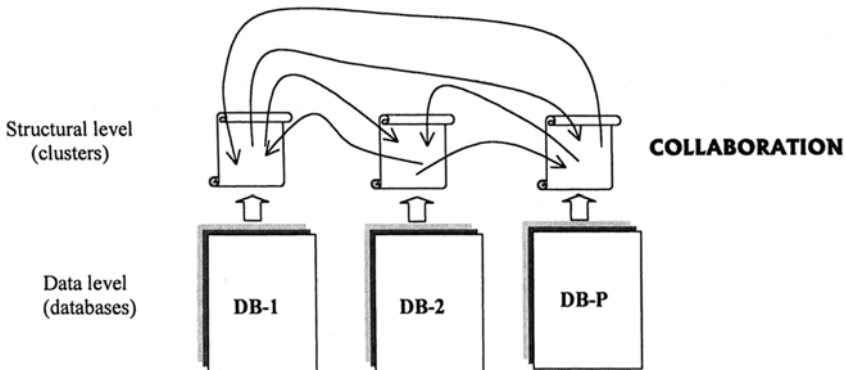


Figure 2. Collaboration between databases at the level of “local” structures (clusters) discovered there; note that no direct collaboration at the data level is allowed.

As data structure elicitation is inherently user-oriented and user-friendly, we are interested in the collaborative clustering as its results are information granules. In the sequel, this gives rise to a certain type of collaboration as indicated before, namely a vertical collaborative clustering that involves databases involving various objects and horizontal clustering where we are faced with the same objects but being characterized by various attributes.

As far as the algorithmic issues are concerned, the underlying idea of collaboration dwells on a well-known Fuzzy C-Means (FCM), see Bezdek (1981) and Duda et al. (2001). As it was stated clearly in our previous investigations, it can be treated as a vehicle of forming information granules. It is also worth stressing that fuzzy clustering arose as a fundamental and highly appealing technique in construction of fuzzy models; refer e.g., to Hopner et al. (1999) and Pedrycz (1998).

First, we proceed with the horizontal collaborative clustering by introducing all necessary notations, formulating the problem itself and discussing its algorithmic aspects. In the sequel, we use a number of numeric examples to illustrate the method. Second, we concentrate on the vertical clustering following the same scheme of presentation as used in the first approach. Illustrative numeric examples are also covered. Next we contrast differences and similarities between vertical and horizontal clustering that are cast in terms of the space in which collaboration occurs. We also raise an issue of data confidentiality and security underling that these terms are well-described in the language of fuzzy sets and become inherently non-Boolean (that is non-binary).

12. 2 THE HORIZONTAL COLLABORATIVE CLUSTERING

Let us start with all necessary notation, formulate the underlying optimization problem implied by the objective function – based clustering technique and derive the solution in a form of some iterative scheme.

The Notation

In what follows, we consider “p” subsets of data located in different spaces (viz. the patterns there are described by different features), $ii=1,2,\dots,p$. As each subset concerns the same patterns (that is each pattern results as a concatenation of the corresponding subpatterns), the number of elements in each subset is the same and equal to N. We are interested in partitioning the data into “c” fuzzy clusters. The result of clustering completed for each subset of data comes in the form of a partition matrix and a collection of prototypes. We use a bracket notation to identify the specific subset. Hence we use the notation $U[ii]$ and $v[ii]$ to denote the partition matrix and the i-th prototype produced by the clustering realized for the ii-th set of data. Similarly, the dimensionality of the patterns (number of their features) in each subset could be different; to underline this we use a pertinent index, say $n[ii]$, $ii=1, 2, \dots, p$. The distance function between the i-th prototype and k-th pattern in the same set is denoted by $d_{ik}^2[ii]$, $i=1, 2, \dots, c$, $k=1, 2, \dots, N$. Again, the index used here (viz. ii) underlines the fact that we are dealing with a certain data space pertinent to the ii-th data set (database). Moreover, we confine ourselves to the weighted Euclidean distance of the form

$$d_{ik}^2[ii] = \|x_k - v_i[ii]\|_{ii} = \sum_{j=1}^{n[ii]} \frac{(x_{kj} - v_{ij}[ii])^2}{\sigma_j^2[ii]} \quad (1)$$

This by no means limits the scope of the approach (that could be easily generalized to any other metric). The objective function guiding the formation of the clusters that is completed for each subset assumes a well-known form as being encountered in the standard FCM algorithm

$$\sum_{k=1}^N \sum_{i=1}^c u_{ik}^2[ii] d_{ik}^2[ii] \quad (2)$$

$ii=1,2, \dots, p$. The collaboration between the subsets is established through a matrix of connections (interaction coefficients or interactions, for brief) , see Figure 3.

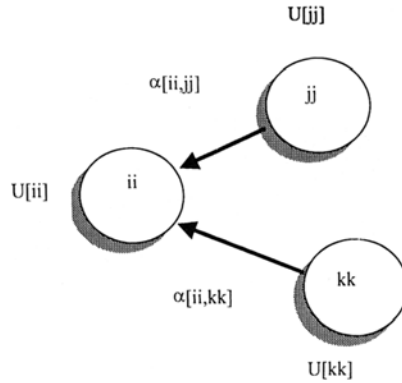


Figure 3. Collaboration in the clustering scheme represented by the matrix of interactions between the data sets.

Each entry of the collaborative matrix states describes an intensity of the interaction. In general, $\alpha[ii, kk]$ assumes nonnegative values. The higher the value of the interaction coefficient, the stronger the collaboration between the corresponding subsets. To accommodate the collaboration effect in the optimization process, the objective function is expanded into the form

$$Q[ii] = \sum_{k=1}^N \sum_{i=1}^c u_{ik}^2[ii] d_{ik}^2[ii] + \sum_{\substack{jj=1 \\ jj \neq ii}}^p \alpha[ii, jj] \sum_{k=1}^N \sum_{i=1}^c \{u_{ik}[ii] - u_{ik}[jj]\}^2 d_{ik}^2[ii] \quad (3)$$

$ii=1, 2, \dots, p$. The role of the second term standing in the above expression is to make the clustering based on the ii -th subset “aware” of the other partitions. It becomes obvious that if the structures in all datasets are similar then the differences between the partition matrices tend to be lower. On the other hand, if we encounter higher differences, we anticipate that the collaboration will be able to address these needs. The weight coefficient ($\alpha[ii, jj]$) help achieve control of the effect of collaboration; the higher the value of $\alpha[ii, jj]$, the more impact comes from the collaboration side of the clustering.

As usual, we require that the partition matrix satisfies a set of “standard” requirements of membership grades summing to 1 for each patterns and the membership grades contained in the unit interval. All in all, the collaborative clustering converts into the following family of “p” optimization problems with membership constraints

$$\begin{aligned} & \text{Min } Q[ii] \\ & \text{subject to} \\ & \quad U[ii] \in \mathbf{U}[ii] \end{aligned}$$

where $\mathbf{U}[ii]$ is a family of all fuzzy partition matrices, namely

$$\mathbf{U}[ii] = \{u_{ik}[ii] \in [0,1] \mid \sum_{i=1}^c u_{ik}[ii] = 1 \text{ for all } k \text{ and } 0 < \sum_{k=1}^N u_{ik}[ii] < N \text{ for } i\}.$$

The minimization is carried out with respect to the fuzzy partition and the prototypes.

From the general point of view, it becomes obvious that the minimization of $Q[ii]$ realized for each granular world separately makes the approach collaborative rather than centralized. The “centralized” version could have been read as follows

$$\text{Min } Q[1] + Q[2] + \dots + Q[p]$$

subject to the constraints conveyed by the partition matrices

$$U[1] \in \mathbf{U}[1], U[2] \in \mathbf{U}[2], \dots, U[p] \in \mathbf{U}[p]$$

where an evident focus is on the additive form of the aggregation of the objective functions.

In the sequel, we proceed with the collaborative clustering.

Optimization Details of the Collaborative Clustering

The above optimization task splits into two problems, namely a determination of the partition matrix $U[ii]$ and the prototypes $v_1[ii]$, $v_2[ii]$, ..., $v_c[ii]$. These problems are solved separately for each of the collaborating subsets of patterns.

To determine the partition matrix, we exploit a technique of Lagrange multipliers so that the constraint occurring in the problem becomes integrated as a part of the objective function considered in the constraint-free optimization. The objective function $V[ii]$ that is considered for each “k” separately comes in the form

$$V[ii] = \sum_{i=1}^c u_{ik}^2[ii] d_{ik}^2[ii] + \sum_{\substack{jj=1 \\ jj \neq ii}}^p \alpha[ii, jj] \sum_{i=1}^c \{u_{ik}[ii] - u_{ik}[jj]\}^2 d_{ik}^2[ii] - \lambda \left(\sum_{i=1}^c u_{ik}[ii] - 1 \right) \quad (4)$$

where λ denotes a Lagrange multiplier. The necessary conditions leading to the local minimum of $V[ii]$ read as follows

$$\frac{\partial V[ii]}{\partial u_{st}[ii]} = 0, \quad \frac{\partial V[ii]}{\partial \lambda} = 0 \quad (5)$$

$s = 1, 2, \dots, c$, $t = 1, 2, \dots, N$. Let us start with the explicit expression governing the optimization of the partition matrix. Computing the derivative of V with respect to u_{st} and zeroing it, we get

$$\frac{\partial V[ii]}{\partial u_{st}[ii]} = 2u_{st}[ii] d_{st}^2[ii] + 2 \sum_{\substack{jj=1 \\ jj \neq ii}}^p \alpha[ii, jj] (u_{st}[ii] - u_{st}[jj]) d_{st}^2[ii] - \lambda = 0 \quad (6)$$

To determine $u_{st}[ii]$, we rewrite this expression as

$$u_{st}[ii] = \frac{\lambda + 2d_{st}^2[ii] \left(\sum_{\substack{jj=1 \\ jj \neq ii}}^p \alpha[ii, jj] u_{st}[jj] \right)}{2d_{st}^2[ii] \left(1 + \sum_{\substack{jj=1 \\ jj \neq ii}}^p \alpha[ii, jj] u_{st}[jj] \right)} \quad (7)$$

To come up with a concise expression, we introduce some auxiliary notation

$$\varphi_{st}[ii] = \sum_{\substack{jj=1 \\ jj \neq ii}}^P \alpha[ii, jj] u_{st}[jj]$$

and

$$\psi[ii] = \sum_{\substack{jj=1 \\ jj \neq ii}}^P \alpha[ii, jj]$$

In virtue of the normalization condition $\sum_{s=1}^c u_{st}[ii] = 1$, the Lagrange multiplier comes as

$$\lambda = \frac{1 - \sum_{s=1}^c \frac{\varphi_{st}[ii]}{1 + \psi[ii]}}{\sum_{s=1}^c \frac{1}{2d_{st}^2[ii](1 + \psi[ii])}}$$

which, taking into account (), leads to the formula

$$u_{st}[ii] = \frac{\varphi_{st}[ii]}{1 + \psi[ii]} + \frac{1}{\sum_{j=1}^c \frac{d_{st}^2}{d_{jt}^2}} \left[1 - \sum_{j=1}^c \frac{\varphi_{jt}[ii]}{1 + \psi[ii]} \right] \quad (8)$$

In the calculations of the prototypes we use explicitly the weighted Euclidean distance between the patterns and the prototypes. The necessary condition for the minimum of the objective function is of the form $\nabla_{v[ii]} Q = 0$. The details are obvious yet the calculations are somewhat tedious. Finally, the resulting prototypes are equal to

$$v_{st}[ii] = \frac{A_{st}[ii] + C_{st}[ii]}{B_s[ii] + D_s[ii]} \quad (9)$$

$s=1, 2, \dots, c, t=1, 2, \dots, n[ii], ii=1, 2, \dots, P$

The coefficients in the above expression are as follows

$$A_{st}[ii] = \sum_{k=1}^N u_{sk}^2[ii] x_{kt}[ii] \quad (10)$$

$$B_s[ii] = \sum_{k=1}^N u_{sk}^2[ii] \quad (11)$$

$$C_{st}[ii] = \sum_{\substack{jj=1 \\ jj \neq ii}}^P \alpha[ii, jj] \sum_{k=1}^N (u_{sk}[ii] - u_{sk}[jj])^2 x_{kt}[ii] \quad (12)$$

$$D_s[ii] = \sum_{\substack{jj=1 \\ jj \neq ii}}^P \alpha[ii, jj] \sum_{k=1}^N (u_{sk}[ii] - u_{sk}[jj])^2 \quad (13)$$

(note that $x_k[ii]$ denotes a k -th pattern coming from the ii -th subset of patterns).

The Detailed Clustering Algorithm: A Flow of Computing

The general clustering scheme consists of two phases:

- generation of clusters without collaboration. This phase involves the use of the FCM algorithm applied individually to each subset of data. Obviously, the number of clusters needs to be the same for all the datasets. During this phase we seek independently a structure in each subset of data
- collaboration of the clusters. Here we start with the already computed partition matrices, set up the collaboration level (through the values of the interaction coefficients arranged in $\alpha[ii, jj]$) and proceed with a simultaneous optimization of the partition matrices

Moving on to the formal algorithm, the computational details are organized in the following way

Given: subsets of patterns $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_P$

Select: distance function, number of clusters (c), termination criterion, and collaboration matrix $\alpha[ii, jj]$.

Initiate randomly all partition matrices $U[1], U[2], \dots, U[p]$

Phase I

For each data

repeat

compute prototypes $\{v_i[ii]\}$, $i=1, 2, \dots, c$ and partition matrices $U[ii]$ for all subsets of patterns

until a termination criterion has been satisfied

Phase II

repeat

For the given matrix of collaborative links $\alpha[ii, jj]$ compute prototypes and partition matrices $U[ii]$ using (4) and (7)

until a termination criterion has been satisfied

The termination criterion relies on the changes to the partition matrices obtained in successive iterations of the clustering method, for instance a Tchebyshev distance could serve as a sound measure of changes in the partition matrices. Subsequently, when this distance is lower than an assumed threshold value ($\epsilon > 0$), the optimization is terminated.

Quantification of the Collaborative Phenomenon of the Clustering

There are two levels of assessing a collaboration effect occurring between the clusters, namely the level of data and the level of information granules (that is fuzzy sets included in the partition matrix). In this latter quantification, we use the results of clustering without any collaboration as a point of reference.

The *level of data* involves a comparison carried out at the level of the numeric representatives of the clustering, that is the prototypes (centroids). The impact of the collaboration is then expressed in the changes of the prototypes occurring as a result of the collaboration.

At the *level of information granules* (partitions and fuzzy sets), the effect of collaboration is expressed in two ways as shown schematically in Figure 4 where the collaboration involves two datasets (viz. $p=2$) indicated by **1** and **2**. Similarly, by **1-ref** and **2-ref** we denote the results (partition matrices) resulting from the clustering carried out without any collaboration. First, we express how close the two partition matrices are as a result of the collaboration. The pertinent measure reads as an average distance between the partition matrices $U_1=[u_{ik}[\mathbf{1}]]$ and $U_2=[u_{ik}[\mathbf{2}]]$, that is

$$\delta = \frac{1}{N * c} \sum_{k=1}^N \sum_{i=1}^c |u_{ik}[\mathbf{1}] - u_{ik}[\mathbf{2}]| \quad (14)$$

Evidently, the stronger the collaboration (higher values of the corresponding α), the lower the values of δ . In this sense, this index helps us translate the collaboration parameters (α) into the effective changes in the membership grades (that are the apparent final result of such interaction). The plot of δ regarded as a function of α is useful in revealing how the collaboration phenomenon takes place. It tells how much the data subset is susceptible to the collaborative impact coming from the other subsets of patterns. For instance, no changes in the values of δ for increasing values of α s is an indicator of strong differences existing between the structures in the two datasets.

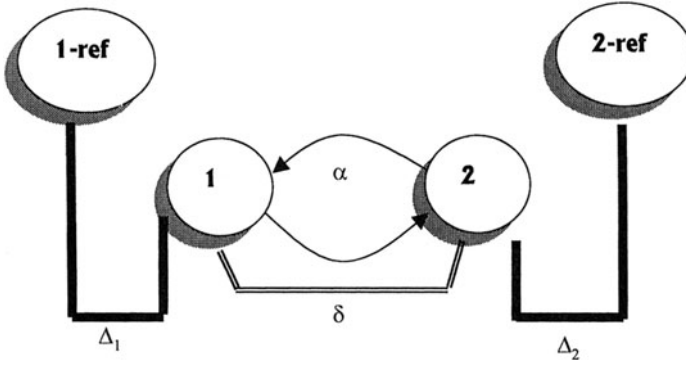


Figure 4. Two ways of quantification of collaboration at the level of information granules; see a detailed description in text.

The second criterion takes into consideration the results of clustering obtained without any collaboration and treats this as a reference point. Using such partition matrices, we quantify how far the collaboration affects the results of clustering. For instance, for the first data set we get

$$\Delta_1 = \frac{1}{N * c} \sum_{k=1}^N \sum_{i=1}^c u_{ik}[1] - u_{ik}[1 - \text{ref}] \quad (15)$$

For the second data subset we obtain

$$\Delta_2 = \frac{1}{N * c} \sum_{k=1}^N \sum_{i=1}^c u_{ik}[2] - u_{ik}[2 - \text{ref}] \quad (16)$$

While the above index exhibit a global character, one can investigate the changes at the level of the individual cluster and patterns. This local behavior of the collaboration is helpful in identifying elements whose membership grades are affected quite significantly as a result of collaboration and those whose structure is compatible across all datasets.

Numerical Examples of Horizontal Collaboration

In the series of numeric experiments, we use a Boston housing data available on the WWW, see <ftp://ftp.ics.uci.edu/pub/machine-learning-databases/housing/>. It consists of 506 patterns describing real estate in the Boston area. There are 14 features describing the patterns. These include crime rate, nitric acid concentration, median

value of the house, just to name a few. We distinguish between two subsets of features where the first one can be treated as descriptors of social aspects of the data

A={per capita crime rate by town, nitric oxides concentration (parts per 10 million), proportion of owner-occupied units built prior to 1940, weighted distances to five Boston employment centers, pupil-teacher ratio by town, % lower status of the population, median value of owner-occupied homes in \$1000's}

and

B={ proportion of residential land zoned for lots over 25,000 sq.ft, proportion of non-retail business acres per town, Charles River dummy variable (equal to 1 if tract bounds river; 0 otherwise), average number of rooms per dwelling, index of accessibility to radial highways, full-value property-tax rate per \$10,000, $1000(B_k - 0.63)^2$ where B_k is the proportion of blacks by town}

In the following experiments we set up the number of the clusters to be equal to 5, $c=5$. Several scenarios of collaboration are discussed; see Figure 5 for its schematic notation. As only two subsets of data are involved, we drop indexes in the collaboration matrix; the meaning of collaboration becomes obvious from the context.

In all experiments we start with clustering that takes place without any collaboration (it was found that the number of iterations equal to 60 was enough to assure no changes to the partition matrices that is the optimization process could be deemed complete). At the next phase the collaboration takes place.

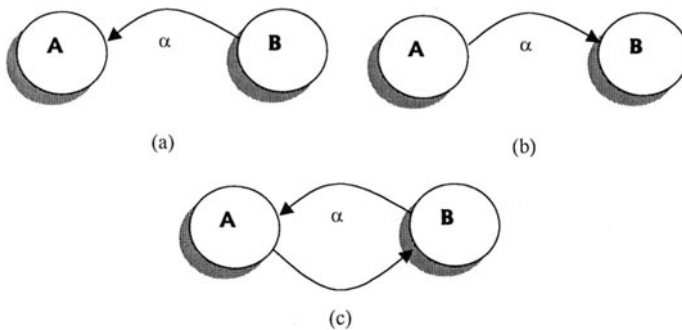


Figure 5. Scenarios of collaborative clustering used in the experiments.

a. There is a collaborative link originating from **B** and affecting **A**. The values of this link (α) are set successively to 0.05, 0.1, 0.5 and 1. The values of the objective function are shown in Figure 5; as expected the objective function assumes higher

values for the increasing levels of collaboration (this is not surprising by noting that the collaboration component contributes additively as a part of this objective function). Noticeable are the drops in the values of the objective function occurring at the beginning of the entire optimization.

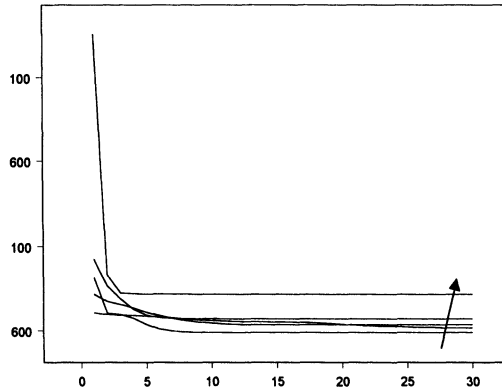


Figure 6. The values of the objective function in successive iteration steps of the algorithm and selected values of the collaborative link (namely 0.05, 0.1, 0.5, and 1).

The resulting prototypes change once the collaboration assumes different intensity as shown below

$\alpha=0.5$

$\mathbf{v}_1 = [12.84 \ 0.68 \ 92.03 \ 1.98 \ 19.90 \ 21.30 \ 13.20]$
 $\mathbf{v}_2 = [0.29 \ 0.44 \ 32.69 \ 6.47 \ 16.88 \ 6.680 \ 27.99]$
 $\mathbf{v}_3 = [0.88 \ 0.53 \ 68.02 \ 3.80 \ 18.73 \ 12.38 \ 21.44]$
 $\mathbf{v}_4 = [0.64 \ 0.50 \ 60.23 \ 4.27 \ 17.78 \ 8.67 \ 26.96]$
 $\mathbf{v}_5 = [8.55 \ 0.66 \ 89.21 \ 2.28 \ 19.96 \ 17.30 \ 17.23]$

$\alpha=1.0$

$\mathbf{v}_1 = [13.19 \ 0.67 \ 91.08 \ 2.02 \ 19.93 \ 20.88 \ 13.25]$
 $\mathbf{v}_2 = [0.26 \ 0.44 \ 33.06 \ 6.54 \ 16.79 \ 6.54 \ 28.45]$
 $\mathbf{v}_3 = [0.76 \ 0.53 \ 67.32 \ 3.85 \ 18.75 \ 12.53 \ 21.19]$
 $\mathbf{v}_4 = [0.56 \ 0.50 \ 59.89 \ 4.34 \ 17.80 \ 8.69 \ 26.75]$
 $\mathbf{v}_5 = [9.42 \ 0.66 \ 88.97 \ 2.24 \ 19.99 \ 17.50 \ 17.11]$

For comparative reasons, the prototypes of the subset **A** without any collaboration are as follows

$v_1 = [11.49 \ 0.69 \ 94.22 \ 1.93 \ 19.94 \ 21.44 \ 13.10]$
 $v_2 = [0.39 \ 0.44 \ 31.90 \ 6.38 \ 17.01 \ 6.96 \ 27.16]$
 $v_3 = [0.86 \ 0.49 \ 52.55 \ 4.61 \ 18.52 \ 9.67 \ 24.04]$
 $v_4 = [1.31 \ 0.54 \ 75.18 \ 3.33 \ 17.24 \ 9.62 \ 27.30]$
 $v_5 = [3.29 \ 0.60 \ 86.47 \ 2.70 \ 19.86 \ 15.53 \ 18.93]$

One can note that higher values of α lead to more evident moves of the prototypes in comparison to their original location when no collaboration took place. The prototypes in each data set start resembling each other. The collaboration effect can be quantified in the language of membership functions (partition matrices). Following the notation introduced in Section 3, the values of the indexes δ and Δ_1 are illustrated in Figure 7.

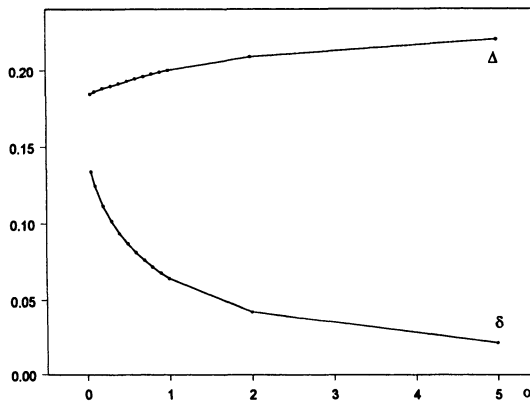


Figure 7. Values of δ and Δ for selected values of α .

As anticipated, the values of δ become lower as the collaboration level increases while Δ_2 gets higher as we depart from the “local” partition matrix (viz. the one computed without any collaboration) being under collaborative pressure to accept some other sources of information about the overall data structure.

b. This experiment deals with the collaboration originating from the second group. The collaboration effect is quantified in Figure 8. In comparison to the other collaborative scheme, there is a quite comparable level of changes in the membership grades. The only significant jump is reported when a collaboration effect comes into a play.

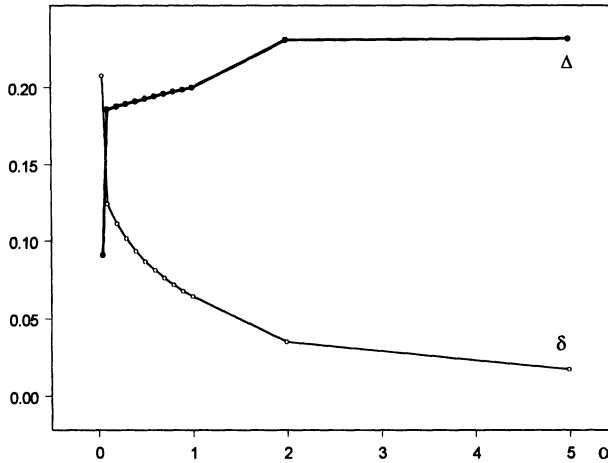


Figure 8. Values of δ and Δ for some selected values of α .

c. In this case, we allow for the collaborative links to be reciprocal that is **A** and **B** interact; see Figure 9. The results are shown in the form of δ as well as Δ_1 and Δ_2 . The values of δ go down monotonically as values of α go higher. An interesting effect occurs in terms of the collaboration: **A** tends to be more stiff as to the collaborative interaction; the values of Δ_1 in spite of the increasing interaction (higher values of α). **B** is more flexible in terms of the collaboration more readily accepting collaborative signals that manifest in the increasing values of Δ_2 .

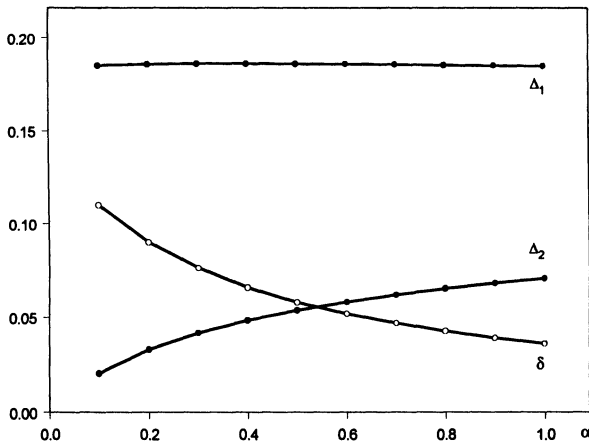


Figure 9 . Values of δ and Δ for selected values of α .

In the following experiments, we split the features into two groups: the first one (**A**) includes all the features but the price of real estate that forms the second group (**B**). The collaborative link is activated by the first group (namely this group affects the clustering realized within **B**).

The prototypes in the median value of house change depending on the values of the collaborative feedback. Noticeably, with the increase of the collaboration (denoted by α) the prototypes tend to occupy more narrow range in comparison to the situation where no interaction was present, see Figure 10.

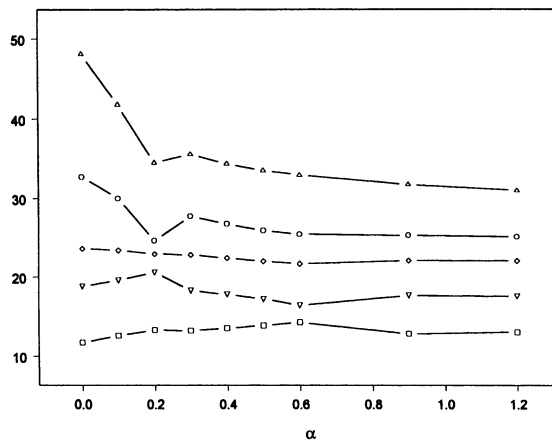


Figure 10. Prototypes in the median value of real state as a function of the collaboration linkage α .

There is also another way of investigating the way of visualizing the effects of collaboration by looking at the changes in the membership grades caused by the collaboration. The changes in the membership grades occurring for the two selected levels of collaboration are shown in Figure 11.

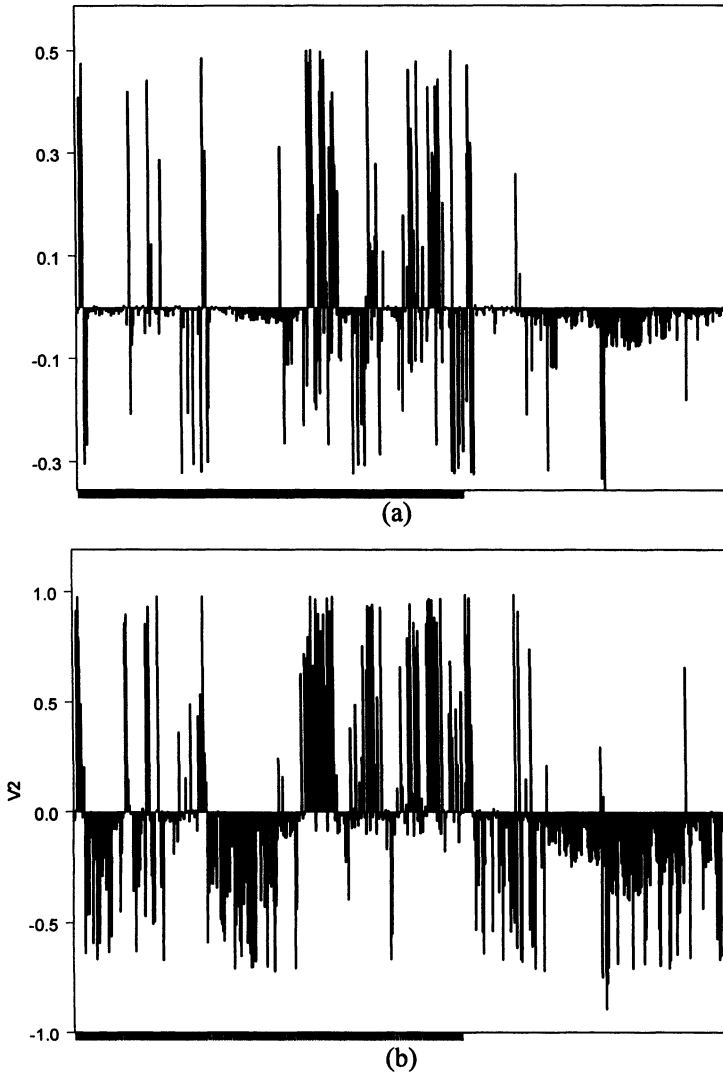


Figure 11. Changes in the membership grades for the first cluster for $\alpha=0.2$ (a) and $\alpha=0.5$ (b).

Now we keep changing the number of clusters while retaining the same level of collaboration ($\alpha = 0.5$) to analyze how this affects the changes of δ and Δ .

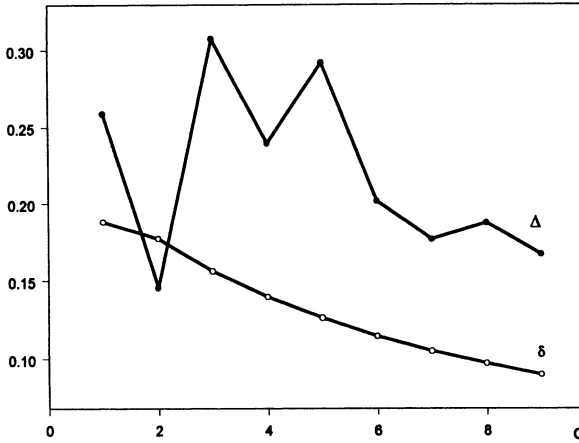


Figure 12. δ and Δ as functions of the number of the clusters (c).

As expected, the values of δ go down with the increasing number of the clusters, Figure 12. The reason for this trend is obvious: we get more clusters, the individual membership grades go down and the differences become lower. As to the second index, it is less monotonic as with the changes of the number of clusters each dataset has its own “plausible” number of clusters and this could vary between them.

12.3 VERTICAL COLLABORATIVE CLUSTERING

As already discussed, the vertical collaborative clustering is concerned with a collection of databases involving different patterns defined in the same feature space so that the patterns do not repeat across the databases. As the feature space is common throughout the databases we can use prototypes as a means of facilitating the collaboration between the databases. The detailed algorithm discussed in the next section concentrates on this form of collaboration.

The Clustering Algorithm

We start with an introduction of the objective function that takes into account the vectors of prototypes specific for each database. With the same notation as before, the objective function is given as

$$Q[ii] = \sum_{i=1}^c \sum_{j=1}^{N[ii]} u_{ik}^2[ii] d_{ik}^2 + \sum_{\substack{jj=1 \\ jj \neq ii}}^P \beta[ii, jj] \sum_{k=1}^{N[ii]} \sum_{i=1}^c u_{ik}^2[ii] \|v_i[ii] - v_i[jj]\|^2 \quad (17)$$

where $\beta[ii, jj] (> 0)$ describes a level of collaboration between the datasets and $\| \cdot \|$ denotes a distance function between the prototypes. The optimization of (15) is carried out for the partition matrix $U[ii]$ and the prototypes of the clusters $v[ii]$. This implies two separate optimization problems where the first one involving the partition matrix is subject to constraints. Not including all computational details, the final expression governing computations of the partition matrix reads in the form

$$u_{st} = \frac{1}{\sum_{j=1}^c \frac{D_{st}^2}{D_{jt}^2}} \quad (18)$$

$t=1, 2, \dots, N[ii], s=1, 2, \dots, c$ where D_{st} is computed as follows

$$D_{st}^2 = d_{st}^2 + \sum_{\substack{jj=1 \\ jj \neq ii}}^p \beta[ii, jj] \| v_s[ii] - v_s[jj] \|^2 \quad (19)$$

Proceeding with the optimization of the prototypes, we express a necessary condition for the minimum of Q to be in the form

$$\frac{\partial Q}{\partial v_s[kk]} = 0, \quad kk = 1, 2, \dots, p \quad (20)$$

This implies a system of linear equations with respect to v_{st} that is

$$v_{st}[ii] = \frac{F_{st}[ii] + A_{st}[ii]}{C_{st}[ii] + B_{st}[ii]} \quad (21)$$

$s=1, 2, \dots, c, t=1, 2, \dots, n$ with the following concise notation

$$\begin{aligned} A_{st}[ii] &= \sum_{k=1}^{N[ii]} u_{sk}^2[ii] x_{kt}[ii] \\ B_{st}[ii] &= \sum_{k=1}^{N[ii]} u_{sk}^2[ii] \\ C_{st}[ii] &= \sum_{\substack{jj=1 \\ jj \neq ii}}^p \beta[ii, jj] \sum_{k=1}^{N[ii]} u_{sk}^2[ii] \\ F_{st}[ii] &= \sum_{\substack{jj=1 \\ jj \neq ii}}^p \beta[ii, jj] \sum_{k=1}^{N[ii]} u_{sk}^2[ii] v_{st}[jj] \end{aligned}$$

The overall computing scheme can be presented in the following fashion

Given: subsets of patterns $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_p$ located in the same feature space

Select: distance function, number of clusters (c), termination criterion, and collaboration matrix $\beta[ii, jj]$.

Initiate randomly all partition matrices $U[1], U[2], \dots, U[p]$

Phase I

For each data

repeat

compute prototypes $\{v_i[ii]\}$, $i=1, 2, \dots, c$ and partition matrices $U[ii]$ for all subsets of patterns

until a termination criterion has been satisfied

Phase II

repeat

For the given matrix of collaborative links $\beta[ii, jj]$ compute prototypes and partition matrices $U[ii]$ using (19) and (16)

until a termination criterion has been satisfied

The vertical collaboration could be realized not only by the prototypes as discussed above but we may establish another vehicle of communication coming in the form of so-called induced partition matrices. The crux of this collaboration is as follows. Consider the prototypes of the clusters located in the data space of the jj -th database, say $v_s[jj]$, $s=1, 2, \dots, c$. Now let us position these prototypes in the data space of the ii -th database. For any element x_t in this data space ($t=1, 2, \dots, N[ii]$), we can compute *induced* membership grades (viz the grades being induced by the prototypes from the different space) in the form

$$u_{st}^{\sim}[ii][jj] = \frac{1}{\sum_{j=1}^c \frac{d_{st}^{\sim}[ii][jj]}{d_{jt}^{\sim}[ii][jj]}} \quad (22)$$

where the distance $\|\cdot\|_{ii}$ is computed in the ii -th space (and this fact is clearly identified by the corresponding subscript), namely

$$d_{st}^{\sim}[ii][jj] = \|x_t - v_s[jj]\|_{ii}^2 \quad (23)$$

Now the objective function for the ii -th dataset can be written down in the form

$$Q = \sum_{i=1}^c \sum_{k=1}^{N[ii]} u_{ik}^2[ii] d_{ik}^2[ii] + \sum_{\substack{jj \\ jj \neq ii}}^p \alpha[ii, jj] \sum_{i=1}^c \sum_{k=1}^{N[ii]} (u_{ik}[ii] - u_{ik}^{\sim}[ii][jj])^2 d_{ik}^2[ii] \quad (24)$$

with the notation already introduced above. The standard optimization requires two steps, that is the calculations of the partition matrix and the prototypes. Let us start

with the partition matrix. Recalling that this implies a constrained optimization, we use Lagrange multipliers that place the standard identity constraint as a part of the objective function, that is

$$V = \sum_{i=1}^c u_{ik}^2[ii] d_{ik}^2[ii] + \sum_{\substack{jj=1 \\ jj \neq ii}}^p \alpha[ii, jj] \sum_{i=1}^c (u_{ik}[ii] - \tilde{u}_{ik}[ii][jj])^2 d_{ik}^2[ii] + \lambda (1 - \sum_{i=1}^c u_{ik}[ii]) \quad (25)$$

for all $k=1, 2, \dots, N[ii]$.

The necessary condition for the minimum of (25) arises in the form

$$\frac{\partial V}{\partial u_{st}[ii]} = 2u_{st}[ii] d_{st}^2[ii] + 2 \sum_{\substack{jj=1 \\ jj \neq ii}}^p \alpha[ii, jj] (u_{st}[ii] - \tilde{u}_{st}[ii][jj]) d_{st}^2 - \lambda = 0$$

Let us introduce the notation

$$D_{st}[ii] = 2d_{st}^2[ii] \left(1 + \sum_{\substack{jj=1 \\ jj \neq ii}}^p \alpha[ii, jj]\right)$$

$$F_{st}[ii] = 2d_{st}^2 \sum_{\substack{jj=1 \\ jj \neq ii}}^p \alpha[ii, jj] \tilde{u}_{st}[ii][jj] d_{st}^2$$

This yields the expression

$$u_{st}[ii] D_{st}[ii] - F_{st}[ii] - \lambda = 0$$

and

$$u_{it}[ii] = \frac{\lambda + F_{it}[ii]}{D_{it}[ii]}$$

that leads to the formula

$$\sum_{i=1}^c \frac{\lambda + F_{it}[ii]}{D_{it}[ii]} = 1$$

Computing the Lagrange multiplier yields

$$\lambda = \frac{1 - \sum_{i=1}^c \frac{F_{it}[ii]}{D_{it}[ii]}}{\sum_{i=1}^c \frac{1}{D_{it}[ii]}}$$

Finally we obtain

$$u_{st}[ii] = \frac{1 - \sum_{i=1}^c \frac{F_{it}[ii]}{D_{it}[ii]}}{\sum_{i=1}^c \frac{D_{st}[ii]}{D_{it}[ii]}} + \frac{F_{st}[ii]}{D_{st}[ii]}$$

$s=1,2, \dots, c, t=1, 2, \dots, N[ii]$.

Proceeding with the computations of the prototypes, we determine them on the basis of the following conditions

$$\frac{\partial Q}{\partial v_{st}[ii]} = 0 \quad (26)$$

that is

$$\begin{aligned} \frac{\partial Q}{\partial v_{st}[ii]} &= \\ &= \frac{\partial}{\partial v_{st}[ii]} \left\{ \sum_{i=1}^c \sum_{k=1}^{N[ii]} u_{ik}^2[ii] \sum_{j=1}^n \frac{(x_{kj}[ii] - v_{ij}[ii])^2}{\sigma_j^2[ii]} \right\} + \\ &+ \frac{\partial}{\partial v_{st}[ii]} \left\{ \sum_{\substack{jj=1 \\ jj \neq ii}}^p \alpha[ii, jj] \sum_{i=1}^c \sum_{k=1}^{N[ii]} (u_{ik}[ii] - \tilde{u}_{ik}[ii][jj])^2 \sum_{j=1}^n \frac{(x_{kj}[ii] - v_{ij}[ii])^2}{\sigma_j^2[ii]} \right\} \\ &= 2 \sum_{k=1}^{N[ii]} u_{sk}^2[ii] \frac{x_{kt}[ii] - v_{st}[ii]}{\sigma_t^2[ii]} + 2 \sum_{\substack{jj=1 \\ jj \neq ii}}^p \alpha[ii, jj] \sum_{k=1}^{N[ii]} (u_{sk}[ii] - \tilde{u}_{sk}[ii][jj])^2 \frac{(x_{kt}[ii] - v_{st}[ii])}{\sigma_t^2[ii]} \end{aligned}$$

To simplify the final formula, let us introduce the notation

$$\begin{aligned} A &= \sum_{k=1}^{N[ii]} u_{sk}^2[ii] x_{kt}[ii] \\ B &= \sum_{k=1}^{N[ii]} u_{sk}^2[ii] \\ C &= \sum_{\substack{jj=1 \\ jj \neq ii}}^p \alpha[ii, jj] \sum_{k=1}^{N[ii]} (u_{sk}[ii] - \tilde{u}_{sk}[ii][jj])^2 x_{kt}[ii] \\ D &= \sum_{\substack{jj=1 \\ jj \neq ii}}^p \alpha[ii, jj] \sum_{k=1}^{N[ii]} (u_{sk}[ii] - \tilde{u}_{sk}[ii][jj])^2 \end{aligned}$$

that combined with (26) produces the final expression

$$v_{st}[ii] = \frac{A + C}{B + D}$$

Let us underline that this type of vertical collaboration occurs in the more abstract space of the information granules (partition matrices) than the previous variant of the collaboration.

Numeric Experiments with Vertical Collaboration

To illustrate how the method of this collaborative clustering works, we consider three collections of two-dimensional synthetic data collected in Table 1 where we identify the indexes of data points in the set. The elements substantially different from one data set to another are indicated in boldface. We partition the data into 3 clusters. The number of iterations the clustering algorithm has been run is equal to 15 (practically, at this number, there are no further changes in the objective function).

No. of data	Data Set #1	Data Set #1	Data Set #1
1	1.1 1.6	1.1 1.6	1.1 1.6
2	1.3 2.1	1.3 2.1	1.3 2.1
3	2.2 2.5	2.2 2.5	2.2 2.5
4	2.3 2.7	2.3 2.7	2.3 2.7
5	3.5 6.7	3.8 8.7	3.8 8.7
6	3.9 6.1	3.9 6.1	3.9 6.1
7	3.3 5.8	5.3 5.8	5.3 5.8
8	2.9 6.2	2.9 6.2	2.9 6.2
9	7.1 9.2	7.1 9.2	5.1 3.2
10	8.3 9.1	8.3 9.1	8.3 3.1
11	7.8 8.5	7.8 5.5	7.8 3.5
12	7.4 7.9	7.4 7.9	2.4 3.9

Table 1. Three datasets used in the experiment of vertical clustering.

For comparative reasons, we start with a scenario in which there is no collaboration. The resulting partition matrices and prototypes are listed below

- partition matrix; first data set

0.962885	0.029110	0.008005
0.987977	0.009657	0.002366
0.973869	0.021538	0.004593
0.948994	0.042514	0.008491
0.010942	0.977999	0.011059
0.013902	0.973516	0.012582
0.010637	0.983763	0.005600
0.015808	0.975499	0.008693
0.007394	0.024593	0.968013
0.006575	0.017805	0.975620
0.000675	0.002018	0.997307
0.009464	0.031015	0.959521

partition matrix; second data set

0.025856	0.964681	0.009462
0.008925	0.988080	0.002995
0.014625	0.981004	0.004371
0.029609	0.961978	0.008414
0.708906	0.075178	0.215916
0.982872	0.008537	0.008591
0.780051	0.067610	0.152339
0.875552	0.078263	0.046185
0.054116	0.012098	0.933786
0.044340	0.012775	0.942884
0.284482	0.093742	0.621776
0.011821	0.002588	0.985591

partition matrix; third data set

0.039863	0.032573	0.927565
0.017405	0.012870	0.969724
0.003631	0.002755	0.993615
0.009680	0.006951	0.983369
0.818173	0.091730	0.090097
0.975737	0.011179	0.013085
0.754797	0.161347	0.083856
0.913704	0.030640	0.055656
0.207793	0.492985	0.299222
0.013797	0.974664	0.011539
0.001866	0.996727	0.001407
0.183367	0.071699	0.744934

In all cases, we have several clearly visible clusters of data. The prototypes of the three datasets as tabulated below, are significantly different. In particular, the second and third prototype varies a lot across the datasets.

Dataset #1- prototypes	Dataset #2 - prototypes	Dataset #3- prototypes
[1.72 2.22]	[1.75 2.25]	[1.91 2.50]
[3.40 6.20]	[4.02 6.49]	[3.87 6.57]
[7.66 8.68]	[7.55 8.29]	[7.66 3.34]

Now, let us set up a collaboration level equal to 1; more specifically, $\beta_{[ii,jj]}=1.0$ for all $ii \neq jj$. The collaboration established in this way results in similar prototypes as quantified in the following table

Dataset #1- prototypes	Dataset #2 - prototypes	Dataset #3- prototypes
[1.87 2.34]	[1.87 2.34]	[1.94 2.40]
[3.72 6.26]	[3.83 6.34]	[3.86 6.24]
[7.44 7.33]	[7.41 7.16]	[7.36 6.14]

Noticeably, the prototypes start exhibiting a strong resemblance across the data that is a visible indicator of the ongoing collaboration. The effect of collaboration driving the prototypes closer for each dataset translates into changes in membership grades of the individual data points. Computationally, the change is taken as the sum of the absolute differences taken over all clusters that is

$$\sum_{i=1}^c |u_{ik} - u_{ik}(\text{no_collaboration})|$$

with $u_{ik}(\text{no_collaboration})$ denoting the membership grade of the k -th pattern in the i -th cluster in case no collaboration is present. This effect of collaboration is shown in Figure xx. Immediately, we recognize that some patterns are quite strongly affected by the collaboration. Those are the patterns that are different between datasets. With the increasing values of β s, the collaboration becomes more vigorous. Subsequently, the values of the changes in the membership grades are shown in Table 2. It can be seen that some of the patterns are heavily affected by the collaboration meaning that at these points the structure are quite distinct and any reconciliation between them requires a substantial level of effort. These particular patterns are indicated in boldface. These results correlate very evidently with the data, see Table 1. The method reveals that data points 9, 10, 11, and 1 are different – an observation included in Table 1. Interestingly, these patterns are the same that resulted in a substantial level of changes in membership occurring during the process of collaboration.

Pattern no.	Change in membership (first dataset)
1	0.032886
2	0.022677
3	0.027390
4	0.043216
5	0.011151
6	0.029918
7	0.064940
8	0.100724
9	0.398732
10	0.323950
11	0.274974
12	0.144609

Pattern no.	Change in membership (second dataset)
1	0.035287
2	0.022392
3	0.014654
4	0.020496
5	0.024933
6	0.012801

7	0.186115
8	0.056257
9	0.366174
10	0.287519
11	0.302670
12	0.186651

Pattern no.	Change in membership (third dataset)
1	0.032886
2	0.022677
3	0.027390
4	0.043216
5	0.011151
6	0.029918
7	0.064940
8	0.100724
9	0.398732
10	0.323950
11	0.274974
12	0.144609

Table 2. Changes in the membership grades of the individual data points in three datasets for $\beta=1.0$.

In the sequel, a total change in the membership (Δ) determined as

$$\Delta = \sum_{k=1}^N \sum_{i=1}^c |u_{ik} - u_{ik}(\text{no_collaboration})|$$

and now regarded as a function of β is summarized in Figure 13. Again, there is a strong monotonic relationship between the level of this collaboration and the manifesting changes in the partition matrix; the detailed relationships vary between datasets (groups of data).

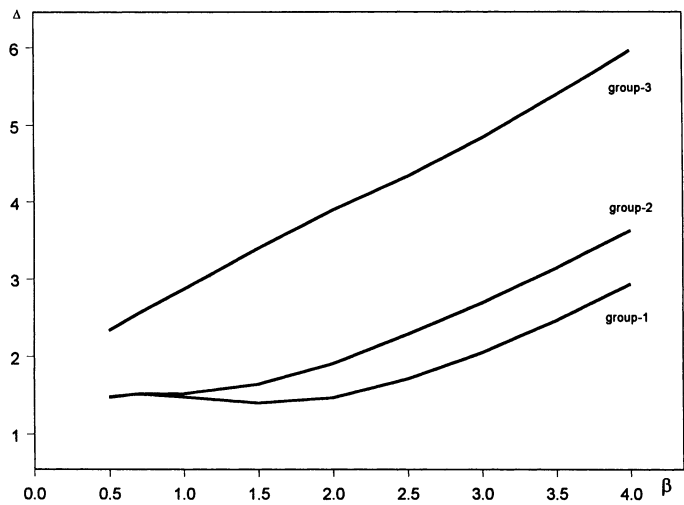


Figure 13. Δ as a function of β for datasets used in the experiment.

Next we consider the same synthetic data now using the vertical clustering where the methods collaborate at the level of the membership functions (partition matrices). First the prototypes obtained at $\beta[i][j]=1$ are summarized in Table 3. Subsequently, we show the changes in the membership grades, Table 4. Finally, Figure 14 visualizes how the values of Δ are affected by the assumed level of the collaboration.

Dataset #1 -prototypes	Dataset #2 -prototypes	Dataset #3-prototypes
[1.82 2.35]	[1.82 2.34]	[2.11 2.53]
[4.12 6.61]	[4.59 6.84]	[4.26 6.17]
[7.63 8.64]	[7.56 8.20]	[7.44 3.62]

Table 3. Prototypes of the clusters for $\beta[i][j]=1$

Dataset-1	Dataset-2	Dataset-3
1 0.041962	1 0.045628	1 0.030457
2 0.031656	2 0.032021	2 0.009927
3 0.037368	3 0.023017	3 0.002122
4 0.065780	4 0.039695	4 0.004779
5 0.105568	5 0.093528	5 0.233347
6 0.044639	6 0.063337	6 0.050150
7 0.296192	7 0.082744	7 0.148840
8 0.340145	8 0.142058	8 0.160036

9 0.486945	9 0.417606	9 0.473936
10 0.424982	10 0.358432	10 0.871235
11 0.410413	11 0.025409	11 0.891050
12 0.371832	12 0.422575	12 0.031298

Table 4. Changes in membership grades for the three datasets; the most significant changes (with the values over 0.3) indicated in boldface.

In general, the patterns that are identified as those requiring a high level of collaboration by the previous method, are also highlighted as such by this approach. Comparing the plots of the changes in Δ , Figure 13 and 14, we see that they are both monotonic yet the type of monotonicity is not identical.

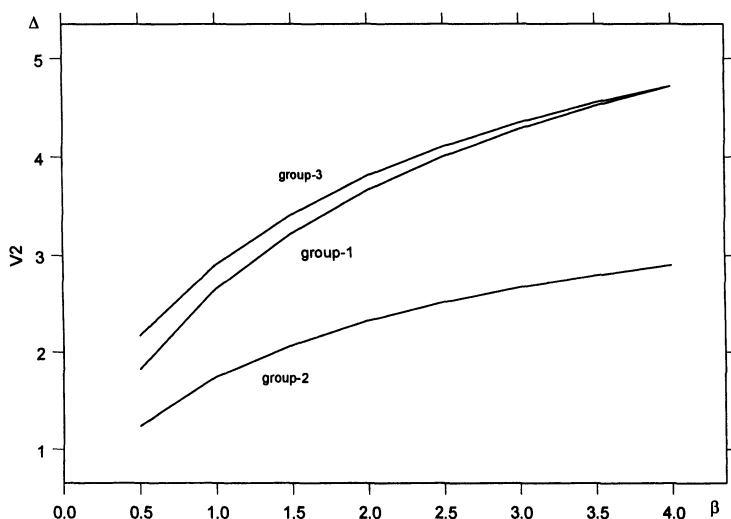


Figure 14. Δ treated as a function of β .

12. 4 VERTICAL AND HORIZONTAL CLUSTERING: COLLABORATION SPACE AND DATA CONFIDENTIALITY AND SECURITY

The two types of collaboration between the clustering methods lead to an interesting taxonomy as far as the space of collaboration is concerned

- in the horizontal collaboration, the methods exchange information in the space of information granules while not being involved in any communication occurring in the data space. We can say that the collaboration happens at more abstract level as the partition matrices being

provided to other collaborators are defined in the space of general data structures (partition matrices). This argument is even more profound by noting that the partition matrices are just collections of fuzzy relations so the collaboration is realized by means of information granules.

- In the vertical collaboration, we encounter two models of collaboration. The first one is the same as outlined in the horizontal collaboration, namely the methods communicate at the level of their partition matrices. This collaboration at the level of information granules that is more abstract than the one going on in the data space itself. The difference lies in the fact that now such partition matrices used for communication purposes are induced ones, namely we compute partition matrices on a basis of the prototypes being available in the data space pertinent to the other dataset. The second model of collaboration is realized in the data space and comes in the form of the data aggregates such as the prototypes. For the induced partition matrices we are concerned with the collaboration of the fuzzy set-based granular worlds. When the prototypes become involved in the collaboration, we arise at the collaboration at the level of the probabilistic information granules (more specifically, we can view prototypes as aggregates of numeric data).

Figure 15 portrays the observations made above in a graphic form by showing details of collaboration realized between the granular worlds.

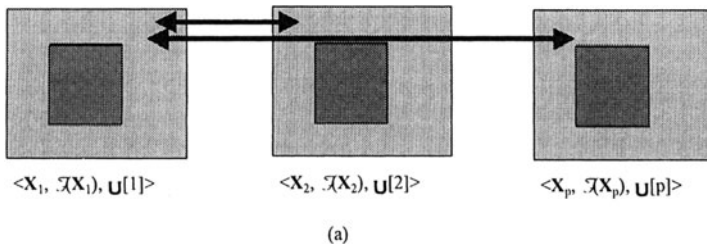


Figure 15(b). Collaborative clustering as an example of communication between granular worlds: horizontal collaboration

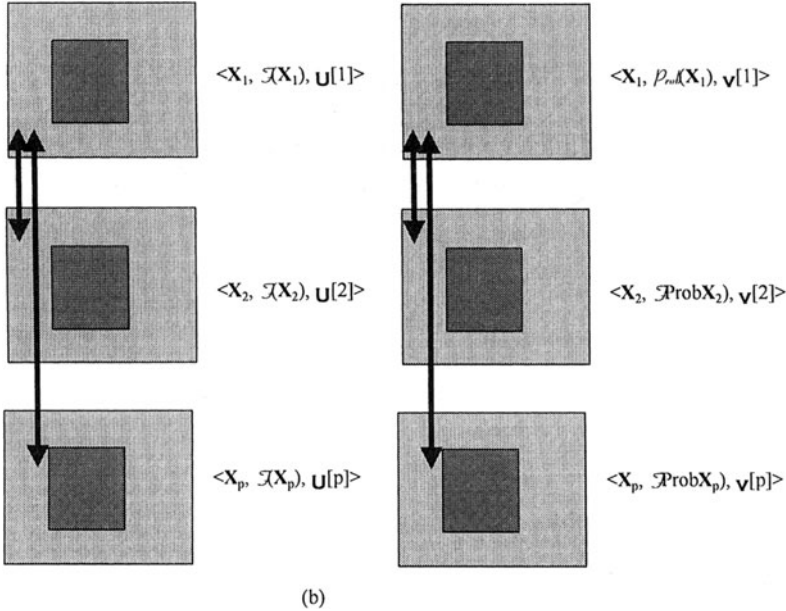


Figure 15(b). Collaborative clustering as an example of communication between granular worlds: vertical collaboration; here we distinguish between two frameworks of granular computing that is built by fuzzy sets $\mathcal{X}(X)$ and probabilistic granules $\mathcal{P}_{\text{rob}}(X)$.

The collaborative computing realized by the clustering methods sheds light on the issue of data confidentiality and security. On one hand, collaboration calls for sharing data. On the other hand, there are some confidentiality requirements that prevent us from a direct access to data in some other databases. The approach taken here is an interesting compromise between these two extremes. We do not *share* data (which is not feasible) but *communicate* at the level of information granules (and this collaboration does not violate the confidentiality requirement). Intuitively, we may note that the larger the information granules (the lower the granularity), the less detailed information we share. This implies that the notion of data confidentiality (and alternatively, data security) is not a Boolean (two-valued) concept but can be represented as a fuzzy set. To maintain a certain level of data confidentiality, we may require that the information transactions (information sharing) occurs at a certain level of information granularity so that we do not get into too detailed information. In particular, we may require that the number of clusters does not exceed a certain maximal value. More formally, we may request that the cardinality

(granularity) of the information granules should not be lower than some threshold value agreed upon during the collaboration contract.

12.5 CONCLUSIONS

We have introduced an idea of collaborative processing, in general and collaborative clustering, in particular. It has been shown that a communication and collaboration between separate datasets can be effectively realized at the more abstract level of membership grades (partition matrices) and prototypes. Two types of collaboration (vertical and horizontal) were studied in detail. We provided a complete clustering algorithm by dwelling the method on the standard FCM method. The quantification of the collaboration effect can be realized either at the level of the prototypes or the partition matrices. An interesting expansion of the method discussed here involves a partial (limited) collaboration where not all patterns are available to form a collaborative link. This simply calls for an extra Boolean vector $\mathbf{b} = [b_1 \ b_2 \ \dots \ b_N]$ modifying the objective function in the form

$$Q[ii] = \sum_{k=1}^N \sum_{i=1}^c u_{ik}^2 [ii] d_{ik}^2 [ii] + \sum_{\substack{jj=1 \\ jj \neq ii}}^P \alpha[ii, jj] \sum_{k=1}^N \sum_{i=1}^c \{u_{ik} [ii] - u_{ik} [jj]\}^2 b_k d_{ik}^2 [ii]$$

where b_k assumes 1 when the k -th pattern is available for collaboration (otherwise b_k is set to 0).

In general, we can envision the collaboration mechanism to take place both at the vertical (data) as well as horizontal (feature) level, see Figure 16. In terms of the objective function, this approach merges the two methods introduced before.

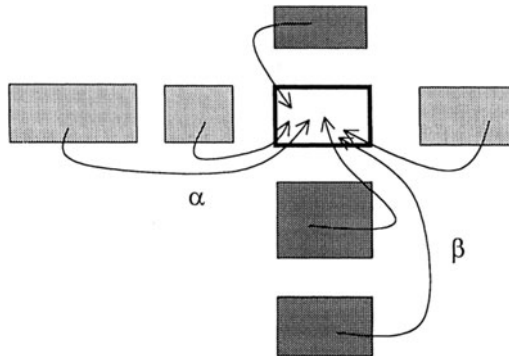


Figure 16. Vertical and horizontal mode of collaboration between datasets.

As a matter of fact, we can put down the following expression to emphasize the collaboration mechanism being in place

$$U[ij]=F(\mathbf{U}[ij], \mathbf{v}[ij])$$

where \mathbf{U} and \mathbf{v} are used to here denote the information feedback of the other part of the system (both in terms of vertical and horizontal collaboration)

The approach presented here could be easily generalized to support more specific ideas such as rule-based systems. In this case, we are concerned with the reconciliation of rules in each subset of data. Obviously, the optimization details need to be refined, as the specificity of the problem requires further in-depth investigations of a number of issues related to rules such as their specificity, consistency and completeness.

REFERENCES

- Agah, A., Tanie, K. (1999), Fuzzy logic controller design utilizing multiple contending software agents. *Fuzzy Sets and Systems*, **106** (2), 121 –130.
- Anderberg, M.R. (1973), *Cluster Analysis for Applications*, Academic Press, N. York.
- Bezdek, J.C. (1981), *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, N. York.
- Duda, R.O., Hart, P.E., Stork, D.K. (2001), *Pattern Classification*, 2nd edition, J. Wiley, N. York, NY.
- Genesereth, M.R., S.P. Ketchpel, S.P. (1994), Software agents, *Communications of the ACM*, **37**(7), 48 -53.
- Hoppner F. et al. (1999), *Fuzzy Cluster Analysis*, J. Wiley, Chichester.
- Loia, V., Sessa, S. (2001), A soft computing framework for adaptive agents. In: *Soft Computing Agents : New Trends for Designing Autonomous Systems*, Springer-Verlag, Heidelberg.
- Pedrycz, W. (1998), Conditional fuzzy clustering in the design of radial basis function neural networks, *IEEE Transactions on Neural Networks*, **9** (4), 601- 612.

DIRECTIONAL MODELS OF GRANULAR COMMUNICATION

13. 1 INTRODUCTION

In the previous chapter, we introduced a model of collaboration between granular worlds. This model was quite general and showed how such collaboration was accomplished in the framework of fuzzy sets by being directly associated with the granular descriptors realized in the form of fuzzy clusters (fuzzy relations). In the proposed mechanisms of collaboration, we have made two assumptions that were consistently kept throughout the entire flow of investigations. First, the collaborative nature of the communication implies that all granular environments are treated in the same manner and there is no particular role being attached to any of them. Second, in virtue of the character of the collaboration, we considered the same number of information granules across all worlds. While these two assumptions are legitimate and apply to a broad range of cases, they can be augmented in several possible ways. The number of the information granules that can exist in each granular world needs not be the same; it is quite evident that the assumption we have made with this regard looks quite restrictive. Furthermore in the collaborative communication we may envision that there are some mappings between some granular worlds and this sense a set of information granules in one space (granular world) may impact the formation of the granules in some other space. The nature of the collaboration is then more *directional* as one of the granular worlds attempts to accommodate some “advice” coming from other environments; the advice is conveyed via some logic-based mapping expressed at the level of information granules. Schematically, we illustrate the crux of this communication in Figure 1. Note that the granular world shown in the center of the figure is influenced by the information granules located at some other worlds. As we have already pointed out, the influence is formulated in the language of some mapping $\Phi[k]$ where “k” underlines a fact that this impact originates from the “k” granular world. As we are now dealing with the mapping, the number of the information granules in each world does not matter. Owing to this directional relationships between the granular worlds, we are referring to the directional type of communication.

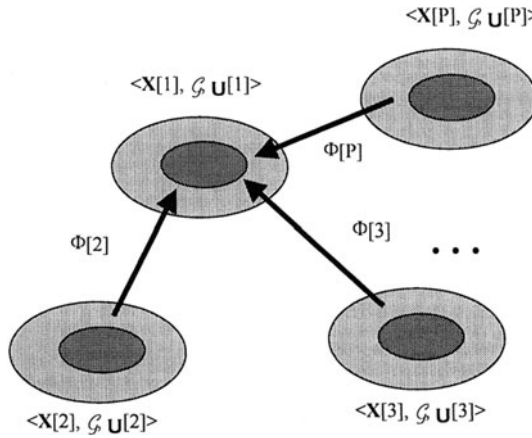


Figure 1. Directional communication between granular worlds; the mapping realized by $\Phi[2]$, $\Phi[3]$, ..., $\Phi[P]$ affect the first granular world.

The detailed directional model of communication is the focal point of this study. We move on with the formulation of a problem that is followed by algorithmic details and numeric illustration. Without any loss of generality, we discuss the directional collaboration with two datasets, $\mathbf{X}[1]$ and $\mathbf{X}[2]$, respectively.

13. 2 PROBLEM FORMULATION

The collaborative way of information granulation leads to the following optimization problem. To formulate it, let us proceed with all necessary notation. To a high extent, we will be alluding to the standard terminology used in fuzzy clustering. Given are two data sets $\mathbf{X}[1]$ and $\mathbf{X}[2]$ where $\mathbf{X}[1] = \{x_1[1], x_2[1], \dots, x_N[1]\}$ and $\mathbf{X}[2] = \{x_1[2], x_2[2], \dots, x_N[2]\}$ where $x_k[1] \in \mathbb{R}^{n_1}$ and $x_k[2] \in \mathbb{R}^{n_2}$. The clustering of $\mathbf{X}[1]$ involves $c[1]$ clusters. For the second data set, the clustering gives rise to $c[2]$ clusters. The clusters constructed in $\mathbf{X}[1]$ are denoted by $A_1, A_2, \dots, A_{c[1]}$. For $\mathbf{X}[2]$ the resulting clusters are $B_1, B_2, \dots, B_{c[2]}$. In what follows, the terms cluster and information granule are used interchangeably. Furthermore, as we are dealing with two data sets, all constructs pertaining to the data are indexed by square brackets, namely [1] and [2].

The mapping between the granules in $\mathbf{X}[1]$ and $\mathbf{X}[2]$ exhibits an evident logic flavor meaning that we assume a logic form of the relationship between the information granules, namely we express an information granule B_i as a logic aggregation (ϕ) involving the information granules developed in $\mathbf{X}[1]$, that is

$$B_i = \varphi(A_1, A_2, \dots, A_{c[1]}, w_i)$$

$i=1, 2, \dots, c[2]$. The single-output mappings (ϕ) defined for all “ i ” are arranged in the multi-input multi-output transformation denoted as Φ .

The above expression includes also a weight vector (parameters) w_i that is used to calibrate the collaborative links between A_j and B_i . We discuss the details of the logic expression in the next section. More descriptively, we are interested in the development of information granules $\{A_i\}$ and $\{B_j\}$ so that they satisfy the requirements of relational and directional nature.

The Objective Function and its Generalization

The entire optimization starts from the objective function defined in such a way that it encapsulates the way of the collaborative formation of the information granules. For illustrative purposes, we start with a specific form of the clustering method that is the well-known FCM algorithm. (this clustering mechanism can be generalized further on when used in this approach). Using the standard notation, the performance index (objective function) assumes the form

✓ for $\mathbf{X}[1]$ the clusters minimize the expression

$$Q[1] = \sum_{i=1}^{c[1]} \sum_{k=1}^N u_{ik}^2[1] d_{ik}^2[1] \quad (1)$$

where

$$d_{ik}^2[1] = \|x_k[1] - v_i[1]\|^2 \quad (2)$$

is a distance function between the pattern (that is $x_k[1]$) and the prototype (denoted here by $v_i[1]$) with both of them being located in $\mathbf{X}[1]$. While the above objective function is well known in the literature, we emphasize that the each row of the partition matrix is just an information granule (more specifically a fuzzy relation) so we can view $U[1]$ in the following form

$$U[1] = [A_1 \ A_2 \ \dots \ A_{c[1]}]^T = \begin{bmatrix} A_1 \\ A_2 \\ \dots \\ A_{c[1]} \end{bmatrix}$$

Note that each A_i is defined in the finite space $\mathbf{X}[1]$ (which implies that A_i comes with a discrete membership function).

The minimization of (1) or equivalently the one provided by (2) is completed with respect to the partition matrix and the prototypes.

- ✓ for $\mathbf{X}[2]$ the clusters built there are affected by the collaboration coming from $\mathbf{X}[1]$ as we are concerned about the mapping (logic transformation between $\mathbf{X}[1]$ and $\mathbf{X}[2]$). This is taken into consideration by expanding the objective function in the following additive form

$$Q[2] = \sum_{i=1}^{c[2]} \sum_{k=1}^N u_{ik}^2[2] d_{ik}^2[2] + \beta \sum_{i=1}^{c[2]} \sum_{k=1}^N (u_{ik}[2] - \phi_i(U[1]))^2 d_{ik}^2[2] \quad (3)$$

The first term is just a standard component encountered in the FCM that looks after the structure in $\mathbf{X}[2]$. The second one captures differences between $U[2]$ and the mapping of the structure detected in $\mathbf{X}[1]$ (viz. the fuzzy partition $U[1]$) to $\mathbf{X}[2]$, that is $\phi_i(U[1])$. It characterizes the performance of the mapping between the information granules. The weight coefficient (β) is used to quantify a balance between the structure in $\mathbf{X}[2]$ and the impact from the mapping requirement (the second term in the above objective function). Considering the two goals of this process of information granulation, we say that β strikes a compromise between the relational and directional aspects of such optimization.

The optimization of the performance index given by (1) proceeds in a standard manner. The optimization of (4) requires detailed investigation. The minimization of the objective function $Q[2]$ is completed with respect to the partition matrix $U[2]$ (structure), prototypes and the parameters of the logic transformation (ϕ_i).

The Logic Transformation

The granular mapping from $\mathbf{X}[1]$ to $\mathbf{X}[2]$ is realized as a logic transformation between the information granules. It is worth stressing that there is a panoply of possible types of mappings and our choice is implied by the transparency of the logic mapping that comes hand in hand with the logic type of the spaces between which the mapping takes place. Two classes of mappings are discussed

- *OR-based* As the name stipulates, we consider the information granule B_i to be an OR aggregation of the granules in the input space, that is

$$B_i = A_1 \text{ or } A_2 \text{ or } \dots \text{ or } A_{c[1]} \quad (4)$$

Not all fuzzy relations in the input space contribute to the formation of B_i nor each of them may exhibit an equal impact on the membership of B_i . To gain this flexibility, we allow for a weight vector (connections) w_i whose role is to articulate (quantify) the contribution coming from A_j 's. The following modification is made to (4)

$$B_i = (A_1 \text{ and } w_{i1}) \text{ or } (A_2 \text{ and } w_{i2}) \text{ or } \dots \text{ or } (A_{c[1]} \text{ and } w_{ic[1]}) \quad (5)$$

where $w_i = [w_{i1} \ w_{i2} \ \dots \ w_{ic[1]}]$ are weights with the values confined to the unit interval. The logic operations are realized by t- and s- norms and this leads to the equivalent expression of (5) (s-t realization of the logic expression)

$$B_i = (A_1 \text{ t } w_{i1}) \text{ s } (A_2 \text{ t } w_{i2}) \text{ s } \dots \text{ s } (A_{c[1]} \text{ t } w_{ic[1]})$$

and

$$B_i = \bigvee_{j=1}^{c[1]} (A_j \text{ t } w_{ij}) \quad (6)$$

Note that the above logic mapping concerns a single fuzzy relation in $\mathbf{X}[2]$. In the similar fashion we can realize the mapping for the remaining information granules. After a careful examination of the mappings being viewed together, we come up with the following concise notation. Arrange all weights into a matrix form

$$R = [r_{ij}] = \begin{bmatrix} w_{11} & \dots & w_{1c[2]} \\ \dots & & \\ & w_{ij} & \\ w_{c[1]1} & \dots & w_{c[1]c[2]} \end{bmatrix} \quad (7)$$

Then the mapping from information granules in $\mathbf{X}[1]$ to information granules in $\mathbf{X}[2]$ is nothing but a fuzzy relational equation with a standard s-t composition (Di Nola et al., 1989; Pedrycz, 1991; Pedrycz and Rocha, 1993; Pedrycz, 1995) (denoted here by a small dot)

$$\mathbf{B} = \mathbf{A} \circ \mathbf{R} \quad (8)$$

where $\mathbf{A} = [A_1 \ A_2 \ \dots \ A_{c[1]}]$ and $\mathbf{B} = [B_1 \ B_2 \ \dots \ B_{c[2]}]$.

Similarly, we can introduce an AND type of aggregation of the information granules meaning that we consider B_i to be a combination of A_j s aggregated AND-wise, that is

$$B_i = A_1 \text{ and } A_2 \text{ and } \dots \text{ and } A_{c[1]} \quad (9)$$

The straightforward generalization of the above aggregation includes a weight vector and, subsequently, the combination of the t-s type

$$B_i = (A_1 \text{ or } w_{i1}) \text{ and } (A_2 \text{ or } w_{i2}) \text{ and } \dots \text{ and } (A_{c[1]} \text{ or } w_{ic[1]}) \quad (10)$$

$$B_i = (A_1 \text{ s } w_{i1}) \text{ t } (A_2 \text{ s } w_{i2}) \text{ t } \dots \text{ t } (A_{c[1]} \text{ s } w_{ic[1]}) = \prod_{j=1}^{c[1]} (A_j \text{ s } w_{ij}) \quad (11)$$

These two aggregation mechanisms are dual in the sense of their functionality. Owing to the character of the AND and OR operations, intuitively we use them depending on the number of the information granules existing in the respective spaces. If $c[1] > c[2]$ we consider the OR type of aggregation (in anticipation that the element in the output space is constructed as a union of several information granules in the input space). Similarly, for $c[1] < c[2]$, the AND-type of aggregation is more appealing (as we project that B_i is made more specific in relation to the information granules existing in $\mathbf{X}[1]$).

13. 3 THE ALGORITHM

Now we are ready to proceed with the computational details that lead us to the complete algorithm. The objective function implies the following optimization task

$$\min_{U[2], v_1[1], v_2[2], \dots, v_{c[2]}} Q[2] \quad (12)$$

subject to

$$U[2] \in \mathbf{U}$$

and

$$R \in \mathbf{R}$$

where the family of partition matrices \mathbf{U} is defined in a usual manner. R is an element of the family of the fuzzy relations \mathbf{R} (viz. matrices with elements confined to the unit interval). The above optimization problem concerns a way of forming a structure in $\mathbf{X}[2]$ with an inclusion of the mapping properties.

The optimization of the partition matrix $U[2]$ in the objective function uses a technique of Lagrange multipliers (because of the constraint existing in the development of the partition matrix). For a given data point (k) , we form an augmented objective function

$$V = \sum_{i=1}^{c[2]} \sum_{k=1}^N u_{ik}^2[2] d_{ik}^2[2] + \beta \sum_{i=1}^{c[2]} \sum_{k=1}^N (u_{ik}[2] - \varphi_i(U[1]))^2 d_{ik}^2[2] + \lambda (\sum_{i=1}^{c[2]} u_{ik}[2] - 1) \quad (13)$$

where λ is a Lagrange multiplier. Proceeding with the necessary conditions for the minimum of V

$$\frac{\partial V}{\partial u_{st}[2]} = 0 \quad \frac{\partial V}{\partial \lambda} = 0$$

we calculate

$$2u_{st}[2]d_{st}^2[2] + 2\beta\beta(u_{st}[2] - y_{st})d_{st}^2[2] + \lambda = 0 \quad (14)$$

where y_{st} stands for a logic-based mapping between the information granules

$$y_{st} = \sum_{j=1}^{c[1]} (u_{st}[1] tr_{sj})$$

Computing $u_{st}[2]$ from (14) we obtain

$$u_{st}[2] = \frac{\beta y_{st} d_{st}^2[2] - \lambda}{2d_{st}^2[2](1 + \beta)}$$

As the membership grades sum up to 1, this leads us to the expression

$$\sum_{j=1}^{c[2]} \frac{\beta y_{jt} d_{jt}^2[2] - \lambda}{2d_{jt}^2[2](1 + \beta)} = 1$$

and in the sequel

$$\frac{\lambda}{1 + \beta} = \frac{-1 + \frac{\beta}{1 + \beta} \sum_{j=1}^{c[2]} y_{jt}}{\sum_{j=1}^{c[2]} \frac{1}{d_{jt}^2[2]}}$$

Introduce the notation

$$\tilde{u}_{st}[2] = \frac{1}{\sum_{j=1}^{c[2]} \frac{d_{st}^2[2]}{d_{jt}^2[2]}}$$

Finally we get

$$u_{st}[2] = \tilde{u}_{st}[2] + \frac{\beta}{1+\beta} (y_{st} - \tilde{u}_{st}[2] \sum_{j=1}^{c[2]} y_{jt}) \quad (15)$$

$s=1, 2, \dots, c[2], t=1, 2, \dots, N.$

The above formula has an interesting interpretation: if β is equal to zero then it reduces to the well-known formula for the partition matrix encountered in the FCM. When β increases, then $u_{st}[2]$ is affected by the second term in (15).

The calculations of the prototypes do not come with any constraints so we follow the necessary condition for the minimum of $Q[2]$, namely $\frac{\partial Q[2]}{\partial v_{st}[2]} = 0, s=1, 2, \dots, c[2], t=1, 2, \dots, n_2.$

In light of the weighted Euclidean distance governed by the expression,

$$d_{ik}^2[2] = \sum_{j=1}^{n_2} \frac{(x_k[j] - v_{ij}[2])^2}{\sigma_j^2[2]} \quad (16)$$

(where $\sigma_j^2[2]$ denotes a variance of the j -th variable), the above derivative is equal to

$$\frac{\partial Q[2]}{\partial v_{st}[2]} = 2\beta \sum_{k=1}^N u_{sk}^2[2] \frac{(x_k[t] - v_{st}[2])}{\sigma_t^2[2]} - 2\beta \sum_{k=1}^N \psi_{sk} \frac{(x_k[t] - v_{st}[2])}{\sigma_t^2[2]} \quad (17)$$

with the following notation

$$\psi_{sk} = (u_{sk}[2] - y_{sk})^2$$

Bearing in mind the necessary condition for the minimum of $Q[2]$ with respect to the prototypes, they are equal to

$$v_{st}[2] = \frac{\sum_{k=1}^N x_k[t] (u_{sk}^2 + \beta \psi_{sk})}{\sum_{k=1}^N (u_{sk}^2 + \beta \psi_{sk})} \quad (18)$$

Noticeably, when $\beta = 0$, we arrive at the standard expression for the prototypes that is identical to the one encountered in the FCM algorithm.

Finally, we optimize the fuzzy relation R describing the logic mapping between the spaces. In general, the solution is not expressed analytically and we have to proceed with some iterative optimization. The underlying expression governing this optimization reads as

$$R(\text{iter}+1) = R(\text{iter}) - \beta \nabla_{R(\text{iter})} Q \quad (19)$$

where the fuzzy relation is transformed on a basis of the gradient of the performance index Q . The learning rate shown above ($\beta > 0$) controls a pace of changes of the updates of the fuzzy relation. The gradient itself is computed for specific triangular norms. In what follows (and all experiments shown in Section 5 will exploit these assumptions) we consider two common models of the logic connectives such as a product (t-norms) and probabilistic sum (s-norm). On this basis, the gradient reads as follows

$$\frac{\partial y_{sk}}{\partial r_{st}} = (1 - A_{st}) u_{tk} [1] \quad (20)$$

where A_{st} denotes an s-t composition that excludes the currently optimized element of the fuzzy relation

$$A_{st} = \sum_{\substack{j=1 \\ j \neq t}}^{c[1]} (u_{jk} [1] \text{tr}_{sj}) \quad (21)$$

13. 4 THE OVERALL DEVELOPMENT FRAMEWORK: A FLOW OF OPTIMIZATION ACTIVITIES

The way in which the information granules are built stipulates a certain flow of optimization activities. These can be grouped into two main phases as outlined in Figure 2. The initial phase concentrates on the clustering completed independently for the two data sets $\mathbf{X}[1]$ and $\mathbf{X}[2]$. The intent here is to establish some preliminary structure in the data so that we could have a reasonable starting point to proceed with the collaboration and further refine the initial relationships. During the second phase, the clustering processes start to collaborate through the mapping. At the same time the fuzzy relation is subject to the gradient-based optimization (as illustrated in Figure 2, this as an integral portion of the collaboration process and negotiation of the granular structures). Because of the direction of the mapping, the clustering in

$\mathbf{X}[1]$ is not affected per se while the relational and directional facets of the clusters emerge at the side of $\mathbf{X}[2]$.

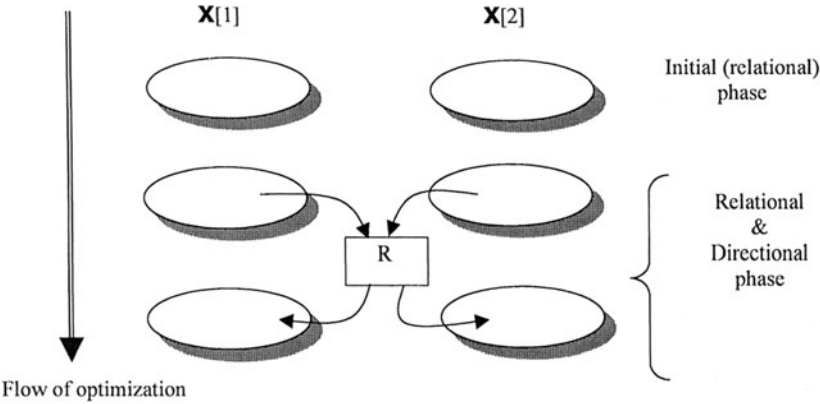


Figure 2. Relational and directional optimization of information granules-an overall development scheme.

13. 5 EXPERIMENTAL STUDIES

The proposed algorithmic framework is illustrated by means of numeric experiments. They include some synthetic as well as real-world data available on the WWW.

Synthetic data The experiment concerns three dimensional data. The first data set includes input variables (x_1 and x_2) while the second involves one-dimensional data (y), refer to Table 1.

x_1	1.2	0.8	0.2	0.9	3.5	4.2	4.3	4.8	6.1	6.5	6.9	6.6	6.4	6.1
x_2	1.8	1.5	1.6	1.2	3.9	3.4	4.7	4.1	7.0	6.2	6.4	5.7	5.8	5.7
y	1.5	1.1	1.4	0.9	3.5	3.2	2.9	3.4	3.6	2.7	2.5	2.8	3.7	3.9

Table 1. Synthetic data used in the experiment.

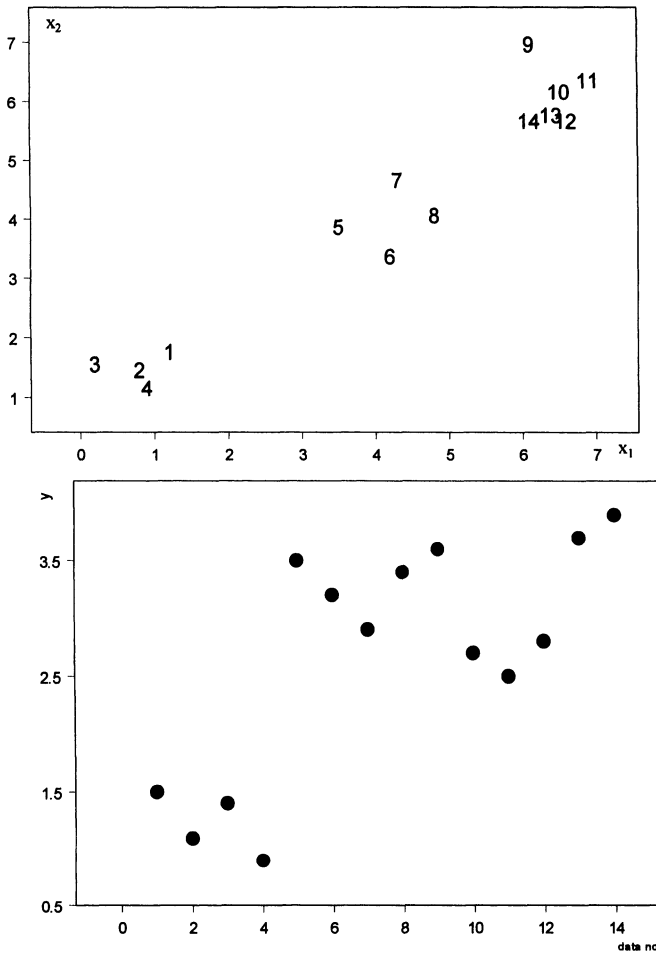


Figure 3. Plots of the synthetic data in the input space (x_1, x_2) and the output data (y).

The same data points are shown in a 3D space in Figure 4. This help reveal the structure. The output variable comes with two clearly visible clusters. Moreover the three clusters in the input space relate to the two clusters in the output space. In more detail, we note that the two clusters in the input space $\mathbf{X}[1]$ map on a single cluster in $\mathbf{X}[2]$.

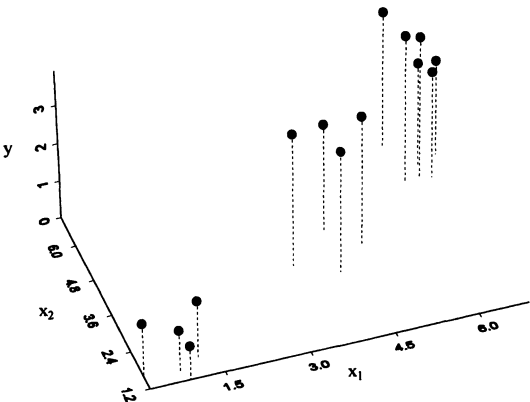


Figure 4. 3D plot of the synthetic data.

Following these observations (which are easy to arrive at as we are dealing with low-dimensional synthetic data), we set up $c[1]=3$ and $c[2]=2$. In the experiment, the learning rate of the gradient-based learning is equal to 0.1. This relatively low value of the learning rate helps avoid oscillations (and this is more crucial to us than an eventual slowdown of the learning process itself). Without any collaboration ($\beta = 0.0$) the obtained clusters are described in terms of the partition matrices:

Partition – space of input variables

$$U[1] = \begin{bmatrix} 0.017434 & 0.005164 & 0.977402 \\ 0.000068 & 0.000022 & 0.999910 \\ 0.013630 & 0.004868 & 0.981502 \\ 0.007460 & 0.002543 & 0.989997 \\ 0.938615 & 0.030536 & 0.030849 \\ 0.936182 & 0.032897 & 0.030921 \\ 0.894294 & 0.082808 & 0.022897 \\ 0.946818 & 0.039856 & 0.013326 \\ 0.069128 & 0.914942 & 0.015930 \\ 0.001119 & 0.998669 & 0.000212 \\ 0.020588 & 0.974971 & 0.004441 \\ 0.027533 & 0.967885 & 0.004582 \\ 0.015457 & 0.982032 & 0.002511 \\ 0.045319 & 0.948161 & 0.006520 \end{bmatrix}$$

Partition – output variable

$$U[2] = \begin{bmatrix} 0.982731 & 0.017269 \\ 0.994259 & 0.005740 \\ 0.994833 & 0.005167 \\ 0.976867 & 0.023133 \\ 0.010273 & 0.989727 \\ 0.001394 & 0.998606 \\ 0.049299 & 0.950701 \\ 0.003565 & 0.996435 \\ 0.019316 & 0.980684 \\ 0.137239 & 0.862761 \\ 0.281143 & 0.718857 \\ 0.086491 & 0.913509 \\ 0.029929 & 0.970070 \\ 0.053702 & 0.946298 \end{bmatrix}$$

Subsequently, the prototypes are equal to

- input space : $v_1[1]=[4.20 \ 4.02]$, $v_2[1]=[6.44 \ 6.12]$, $v_3[1]=[0.78 \ 1.52]$
- output space $v_1[2]=1.27$ $v_2[2]=3.27$

The clusters emerging in both spaces are very well delineated with a very limited overlap.

The collaborative clustering is carried out for several levels of collaboration (β). Following the general scheme (see Section 4), we implement the collaboration after the initial clustering of the individual data. Here we go for 5 iterations. The performance index achieved throughout the clustering and learning of the relations is shown in Figure 5 (note that the term “cycle” used there concerns the performance index recorded for a single clustering iteration and 20 learning epochs of the gradient-based learning). The optimization is efficient as the values of the performance index are reduced from cycle to cycle.

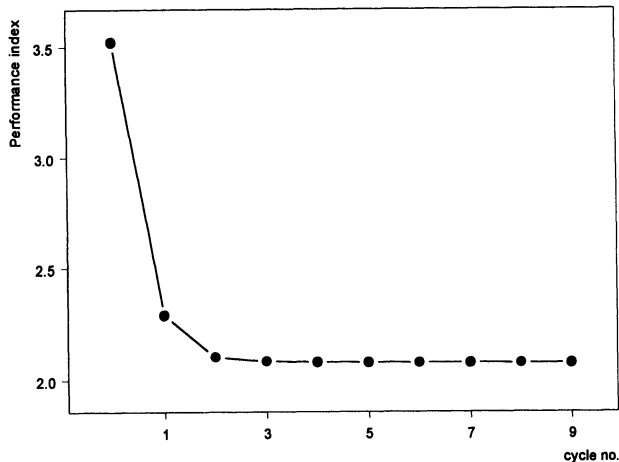


Figure 5. Performance index in successive development cycles ($\beta = 0.1, \alpha = 0.1$).

Once the optimization has been completed, the fuzzy partition in the output space is as presented below

$$U[2] = \begin{bmatrix} 0.980664 & 0.019336 \\ 0.994456 & 0.005544 \\ 0.993309 & 0.006691 \\ 0.976525 & 0.023475 \\ 0.010258 & 0.989742 \\ 0.001312 & 0.998688 \\ 0.045444 & 0.954556 \\ 0.003853 & 0.996147 \\ 0.023591 & 0.976409 \\ 0.128656 & 0.871345 \\ 0.259470 & 0.740530 \\ 0.082898 & 0.917102 \\ 0.033662 & 0.966339 \\ 0.055437 & 0.944563 \end{bmatrix}$$

We do not report the results of clustering in the input space as in this model these fuzzy sets have not been affected. When comparing this partition matrix with the one obtained for the clustering without any collaboration, we conclude that there are no

substantial differences. Obviously, the collaboration effect is quite limited and this may be a reason behind an evident coincidence in the information granules (being conveyed in the respective partition matrices). The prototypes do not change when the collaboration effect comes into the play at this level (namely for $\beta = 0.1$).

$$U[2] = \begin{bmatrix} 0.980664 & 0.019336 \\ 0.994456 & 0.005544 \\ 0.993309 & 0.006691 \\ 0.976525 & 0.023475 \\ 0.010258 & 0.989742 \\ 0.001312 & 0.998688 \\ 0.045444 & 0.954556 \\ 0.003853 & 0.996147 \\ 0.023591 & 0.976409 \\ 0.128656 & 0.871345 \\ 0.259470 & 0.740530 \\ 0.082898 & 0.917102 \\ 0.033662 & 0.966339 \\ 0.055437 & 0.944563 \end{bmatrix}$$

What becomes of interest is a fuzzy relation revealing main relationships between the information granules (fuzzy sets) in the input and output spaces,

$$R = \begin{bmatrix} 0.000000 & 0.063547 & 0.738940 \\ 0.710587 & 0.889359 & 0.005974 \end{bmatrix}$$

There is a strong dependency (relationship) between the granules quantified by high membership grades. Denoting the fuzzy sets by A_1 , A_2 , and A_3 (input space) and B_1 and B_2 (output space), we translate the above fuzzy relation into two logic expressions

$$\begin{aligned} B_1 &= A_3 (0.73) \\ B_2 &= A_1 (0.71) \text{ or } A_2 (0.89) \end{aligned}$$

(note that we have included only the terms with high levels of association; the associations themselves are simply the corresponding entries of the fuzzy relation)

Now we increase the collaboration level to 0.4. This results in the partition matrix whose entries start to divert from the ones without any collaboration. Figure 6 illustrates these new membership grades of the partition matrices.

membership

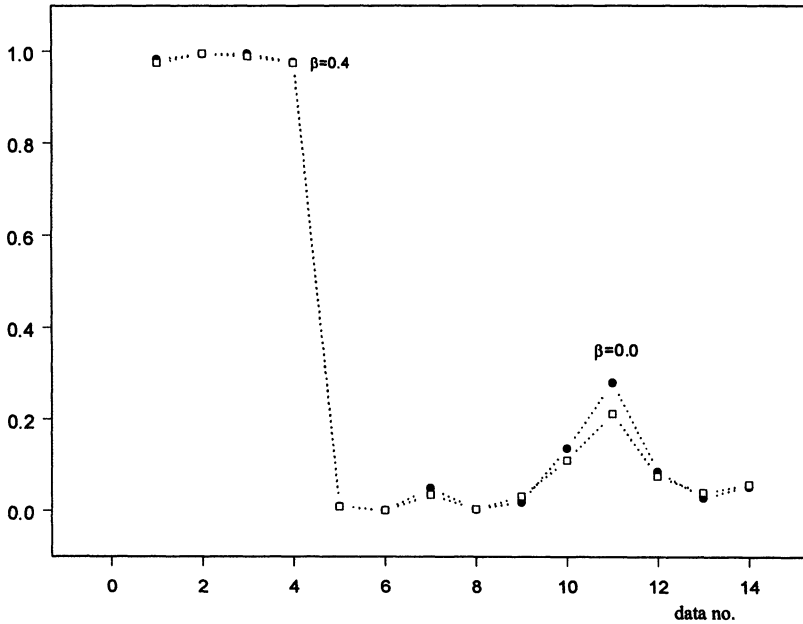


Figure 6. Changes in membership grades as a result of collaboration.

The differences between the prototypes are still negligible as they are equal to $v_1[2]=1.25503$ and $v_2[2]=3.255817$, respectively. The fuzzy relation of the associations is now equal to

$$R = \begin{bmatrix} 0.000000 & 0.060559 & 0.947464 \\ 1.000000 & 0.908921 & 0.005793 \end{bmatrix}$$

Subsequently, the list of logical expressions is similar to the one obtained before

$$\begin{aligned} B_1 &= A_3 (0.95) \\ B_2 &= A_1 (1.00) \text{ or } A_2 (0.91) \end{aligned}$$

however now the strength of the associations between the information granules has been elevated.

Noticeably, higher values of β may lead to instability as the mechanisms used in the method tend to “compete”. This is visible in Figure 7: for higher β , the performance index exhibits some tendency to oscillate. These tend to become more visible once the structures tend to rely on each other more significantly (β increases). The lack of stability points out that we are now faced with a sort of competition between the structures as they do not collaborate any longer but tend to compete.

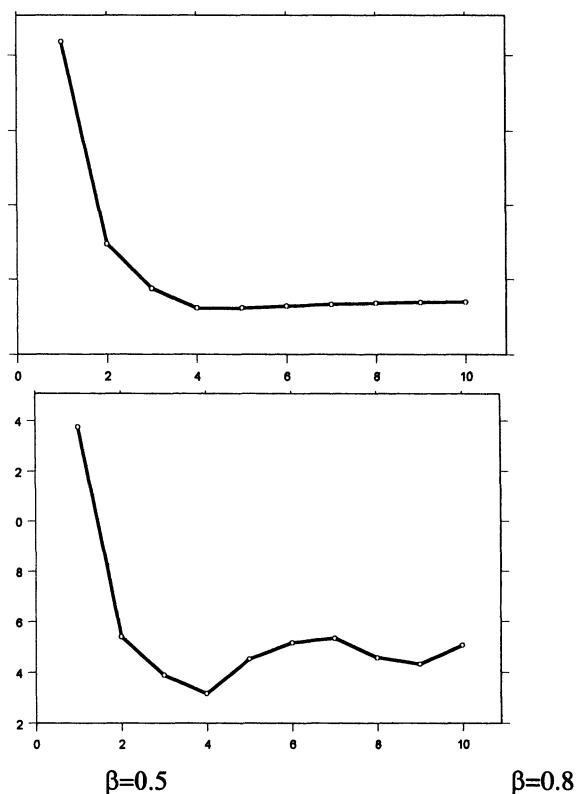


Figure 7. Performance index $Q[2]$ in successive cycles of optimization for two selected values of β .

Auto-mpg dataset comes from the UCI repository of machine learning (<http://www.ics.uci.edu/~mllearn/MLSummary.html>) and concerns a collection of vehicles described in terms of their displacement, weight, country of origin, etc. We consider all features but the fuel efficiency (expressed in miles per gallon) as inputs. The fuel efficiency is treated as the output variable.

The clustering is carried out for different number of the clusters in the input and output space. The level of collaboration (β) is maximized as much as the stability is retained. The results are summarized in the form of the fuzzy relations (with the most essential links being highlighted) and the prototypes in the input and output spaces. Table 2 contains a sample of the findings. Noticeably, there are no significant changes to the information granules. The granules in the input space start to become more specific once its number increases.

$$\beta=0.1$$

$$R = \begin{bmatrix} 0.000000 & \mathbf{1.000000} & \mathbf{0.623695} & 0.000000 \\ \mathbf{0.853638} & 0.000000 & 0.000000 & 0.359311 \end{bmatrix}$$

prototypes

input space

$$\begin{aligned} v_1[1] &= [4.07 \ 103.92 \ 76.34 \ 2219.61 \ 16.55 \ 77.51 \ 2.66] \\ v_2[1] &= [7.93 \ 347.37 \ 160.88 \ 4164.81 \ 12.66 \ 73.34 \ 1.01] \\ v_3[1] &= [5.87 \ 218.63 \ 99.78 \ 3196.26 \ 16.62 \ 75.81 \ 1.11] \\ v_4[1] &= [4.22 \ 127.185 \ 83.87 \ 2475.72 \ 16.14 \ 77.31 \ 1.42] \end{aligned}$$

output space

$$v_1[2] = 17.54 \quad v_2[2] = 31.19$$

$$\beta=0.15$$

$$R = \begin{bmatrix} \mathbf{1.000000} & 0.000000 & \mathbf{0.509261} & 0.000000 & 0.020741 \\ \mathbf{0.628279} & \mathbf{1.000000} & 0.305735 & \mathbf{0.521905} & 0.353167 \end{bmatrix}$$

prototypes

input space

$$\begin{aligned} v_1[1] &= [4.07 \ 102.47 \ 73.39 \ 2188.47 \ 16.73 \ 78.91 \ 2.79] \\ v_2[1] &= [7.96 \ 350.07 \ 162.00 \ 4189.60 \ 12.61 \ 73.22 \ 1.00] \\ v_3[1] &= [4.27 \ 137.07 \ 85.29 \ 2578.33 \ 16.20 \ 79.02 \ 1.18] \end{aligned}$$

$v_4[1] = [6.07 \ 230.51 \ 102.26 \ 3290.02 \ 16.56 \ 75.63 \ 1.06]$
 $v_5[1] = [4.15 \ 113.29 \ 84.53 \ 2359.72 \ 15.94 \ 74.02 \ 2.03]$

output space

$v_1[2] = 31.105927$ $v_2[2] = 17.784975$

$\beta = 0.2$

$R = \begin{bmatrix} 0.209803 & 0.261901 & 0.032446 & 0.000000 & 0.010396 & \mathbf{0.751416} & \mathbf{1.000000} \\ 0.224672 & \mathbf{1.000000} & \mathbf{1.000000} & \mathbf{0.623729} & 0.181158 & 0.135858 & 0.318719 \end{bmatrix}$

prototypes

input space

$v_1[1] = [4.09 \ 108.24 \ 83.89 \ 2309.29 \ 15.86 \ 73.81 \ 2.21]$
 $v_2[1] = [7.96 \ 362.46 \ 168.89 \ 4264.13 \ 12.25 \ 72.35 \ 1.00]$
 $v_3[1] = [7.79 \ 309.14 \ 138.76 \ 3885.15 \ 13.94 \ 76.40 \ 1.02]$
 $v_4[1] = [5.95 \ 226.26 \ 99.46 \ 3241.92 \ 16.69 \ 75.18 \ 1.05]$
 $v_5[1] = [4.33 \ 130.68 \ 84.46 \ 2507.56 \ 16.51 \ 76.25 \ 1.50]$
 $v_6[1] = [4.21 \ 135.32 \ 84.60 \ 2569.33 \ 16.17 \ 79.52 \ 1.17]$
 $v_7[1] = [4.04 \ 99.76 \ 71.44 \ 2154.74 \ 16.82 \ 79.41 \ 2.87]$

output space

$v_1[2] = 31.248861$ $v_2[2] = 17.654987$

Table 2. Results of collaboration between clusters in the input and output spaces for selected number of the clusters in the input space, $c[1] = 4, 5, \text{ and } 7$ and $c[2] = 2$. The table shows also the maximal values of the collaboration factors (β) and the obtained prototypes of the clusters.

It is interesting to note that the collaboration can be made more vigorous without scarifying stability when the number of the clusters in the input space increases. This could have been expected, as the resulting information granules tend to be smaller (of higher granularity) and therefore could be moved around more freely not causing too much distortions (and hence instabilities) during the collaboration process. The dependencies between the information granules as expressed by the fuzzy relations discriminate quite well between strong and weak links. In other words, the fuzzy relations start to contain values either close to 0 or 1. This points out that some information granules relate very strongly.

Now let us consider the same number of the clusters in both spaces, Table 3. This arrangement helps us reveal how the granules relate in the two spaces. Because of the same number of the fuzzy sets, the logic formula may be of a form of one-to-one mapping, namely a mapping a single information granule in the input space to some other information granule in the output space. Obviously, this happens at the level of information granules rather than numeric quantities. By considering the entries of the fuzzy relations, this observation about this one-to-one is fully legitimate. In each row of the fuzzy relation, we have only one dominant membership grade (indicated in boldface in Table 3). Noticeably, these are not necessarily high membership values. This is, however, justified as the partition matrices start having lower entries once the number of the clusters goes up (recall that these membership grades have to add up to 1).

$$R = \begin{bmatrix} 0.000000 & \mathbf{0.949054} & 0.000000 \\ \mathbf{0.420400} & 0.000000 & 0.000000 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.110320 & 0.025040 & \mathbf{0.247469} \\ 0.000000 & \mathbf{0.877086} & 0.000000 \\ \mathbf{0.210435} & 0.000000 & 0.000000 \end{bmatrix}$$

$$R = \begin{bmatrix} \mathbf{0.090541} & 0.003166 & 0.000000 & 0.000788 & 0.000000 & 0.003596 \\ 0.004518 & 0.012588 & 0.076320 & 0.017965 & \mathbf{0.106748} & 0.006710 \\ 0.000000 & 0.051439 & \mathbf{0.156852} & 0.109466 & 0.000000 & 0.000000 \\ \mathbf{0.160533} & 0.000564 & 0.000000 & 0.000000 & 0.000000 & 0.037163 \\ 0.031722 & 0.004176 & 0.000000 & 0.003358 & \mathbf{0.117081} & 0.075881 \\ 0.000000 & \mathbf{0.706356} & 0.000000 & 0.000000 & 0.000000 & 0.000000 \end{bmatrix}$$

Table 3. Fuzzy relations of connections for $c[1]=c[2]=2, 3$, and 6 with $\beta=0.05$.

The graph of links between the information granules for $c=2$ and 3 are included in Figure 8 (we show only the most dominant connections).

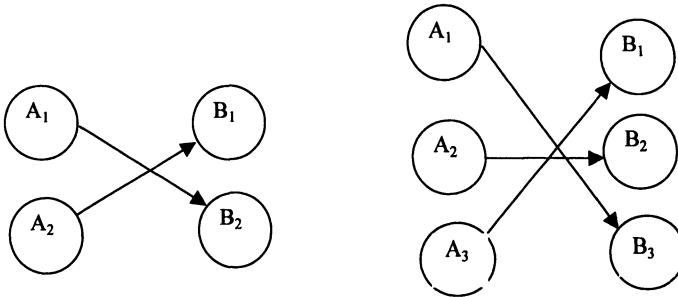


Figure 8. A graphical illustration of linkages between information granules in $\mathbf{X}[1]$ and $\mathbf{X}[2]$ (the most significant connections included).

13. 6 CONCLUSIONS

In this study, we raised an issue of designing information granules (fuzzy sets or fuzzy relations) that takes into consideration a structure in a data set as well as addresses the mapping aspects occurring at the level of such information granules.

The essential novelty of this approach resides with this multifaceted aspect of information granulation. Fuzzy clustering itself is after the structure of the data and does not look into the nature of possible mappings. This is evident that fuzzy clustering no matter which technique we study, tackles a *relational* nature of data (so no direction when searching for a structure is taken into consideration). The augmented objective function includes an additional collaboration component to make the information granules in rapport with the mapping requirements (that comes with a *directional* component). The additive form of the objective function with a modifiable component of collaborative activities makes it possible to model the level of collaboration and avoid a phenomenon of potential competition in case of incompatible structures and the associated mapping.

The logic-based type of mapping (that invokes the use of fuzzy relational equations) comes as a consequence of the logic framework of information granules. One can, however, apply other types of mapping including those implemented via neural networks. This generalizes the approach and promotes it as a general model of collaborative granular computing.

The collaborative scheme of information granulation is also in line with a broad range of techniques of fuzzy modeling regarded as a special category of granular modeling, cf. Delgado et al. (1997), Ma et al. (2000), Pedrycz and Vasilakos (1999), Setnes (2000), Sugeno and Yasukawa (1993), Zhang and Kandel (1998). The agenda of fuzzy modeling is to build transparent models operating at the level of

information granules viewed as fuzzy sets or fuzzy relations. At the operational end, the construction of the tangible and semantically sound information granules becomes crucial to the integrity of any fuzzy model. Furthermore the agenda of fuzzy modeling includes an important aspect of accuracy of the model. This calls for the adjustment of the fuzzy sets quite often to the point where their modifications tend to be difficult to justify from the standpoint of interpretability of the model. The point is in the construction of the information granules so that they maintain their semantics as well as contribute to the accuracy of the model. In essence, there are two requirements to meet. The collaborative form of the fuzzy clustering is a suitable framework of fuzzy modeling in which these two needs are fulfilled in unison.

REFERENCES

- Bargiela, A., Pedrycz, W. (2001), Granular clustering with partial supervision, *European Simulation Multiconference ESM2001*, Prague, June 2001, 113-120.
- Bezdek, J.C. (1981), *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, N. York.
- Di Nola, A., S. Sessa, S., Pedrycz, W., E. Sanchez, E.(1989), *Fuzzy Relational Equations and Their Applications in Knowledge Engineering*, Kluwer Academic Press, Dordrecht.
- Delgado, M., Gomez-Skarmeta, A. F., Martin, F.(1997), A fuzzy clustering-based prototyping for fuzzy rule-based modeling, *IEEE Transactions on Fuzzy Systems*, **5**(2), 223-233.
- Ma, M., Zhang, Y.-Q., Langholz, G., Kandel, A. (2000), On direct construction of fuzzy systems, *Fuzzy Sets and Systems*, **112**, 165-171.
- Pedrycz, W.(1991), Neurocomputations in relational systems, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **13**, 289-296.
- Pedrycz, W., Rocha, A. (1993), Knowledge-based neural networks, *IEEE Trans. on Fuzzy Systems*, **1**, 254-266.
- Pedrycz, W. (1995), *Fuzzy Sets Engineering*, CRC Press, Boca Raton, Fl.
- Pedrycz, W., Vasilakos, A.V. (1999), Linguistic models and linguistic modeling, *IEEE Trans. on Systems Man and Cybernetics*, 1999, **29**(6), 745-757.
- Setnes, M. (2000), Supervised fuzzy clustering for rule extraction, *IEEE Transactions on Fuzzy Systems*, **8**(4), 2000, 416-424.
- Sugeno, M., Yasukawa, T. (1993), A Fuzzy-logic-based approach to qualitative modeling, *IEEE Transactions on Fuzzy Systems*, **1**(1), 7-31.
- Zhang, Y.-Q., Kandel, A. (1998), *Compensatory Genetic Fuzzy Neural Networks and Their Applications*, World Scientific, Singapore.

INTELLIGENT AGENTS AND GRANULAR WORLDS

In the previous chapters, we discussed the principle of communication between granular worlds. We have not looked at the architectural and algorithmic details of these worlds. This is the objective of this chapter. We get into detail of intelligent agents embedded in the corresponding granular worlds. A general topology of a finite state machine and more specifically fuzzy finite state machine is discussed as a comprehensive computing model. These models are particularly appealing because of their memory-based processing (so we are really concerned with dynamic systems). The underlying design process immensely benefits from available learning schemes.

14. 1 INTRODUCTION

Agent technologies are rapidly growing area of information technology. In spite of some differences, the standard definitions of agents emphasize their autonomous nature, and learning abilities (evolving behavior), cf. Kafura and Briot (1998), Cerri (1997). Some other descriptions underline an aspect of communication between the agents, which is regarded as an important facet of any collective activity pertinent to this area (Osman, Bargiela, 2000; Wagealla, Osman, Bargiela, 2002). A comprehensive and lucid discussion on intelligent agents developed in a fuzzy evolutionary framework can be found in Ciscialese et al. (1999). By looking into the specificity of the agent-based activities, it is not surprising that a technology of fuzzy set has a lot to offer. First, a distributed problem solving is completed at various levels of generality. Second, agents collaborate and communicate between themselves, (refer to Figure 1). Fuzzy sets are one among several key vehicles of granular computing. Fuzzy sets themselves are examples of information granules. Depending upon the level of granularity, various communication links can be established. Two agents can solve the problem at a certain level of detail. They could collaborate efficiently if the level of granularity of these agents is similar. Agents are dynamic systems: accept inputs and generate outputs depending on its internal state.

In this sense, the internal dynamic structure of the agents is an important feature of the autonomous agent.

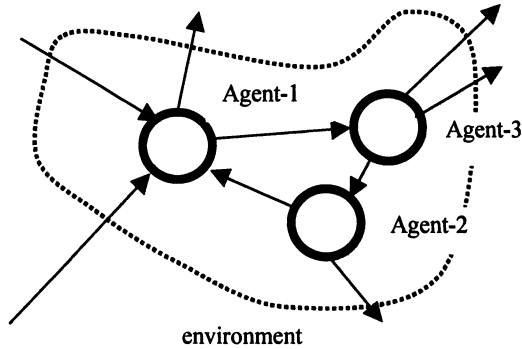


Figure 1. An environment of agent-based computing; note a distributed character of problem solving and an important role of communication and interaction between the agents.

First, we revisit the concept of autonomous agents in the setting of fuzzy sets with a special emphasis paid to the communication problems in the collective environment formed by the agents and the development of a formal model of the agent itself. We employ a top-down approach starting from an overall architecture and then proceeding with a detailed model of the agent. First, we concentrate on the problem of communication between agents, introduce a notion of a communication quality and quantify it numerically through a suitable communication index. Next, we discuss on a concept of a fuzzy state machine (fuzzy automaton) regarded as a generic model of an agent by concentrating on its conceptual and learning capabilities. A fuzzy flip-flip (more precisely a fuzzy JK flip-flop) is a basic building block of the state machines and look more thoroughly at its dynamics. Subsequently, we provide with a detailed learning scheme for developing fuzzy state machines.

14. 2 COMMUNICATION BETWEEN THE AGENTS IN THE GRANULAR ENVIRONMENT

Apparently, agents are autonomous systems. There is no central control (coordination centre) but agents cooperate in representing and solving problems. Each agent may have its specific agenda that looks into problem solving from a very unique standpoint. Communication mechanisms or more generally, a communication environment in which they operate is of primordial importance. This concerns both the communication links to be established between the agents as well as the external

environment with which they interact. The following communication aspects deserve careful analysis

- An interaction realized in terms of very different (and therefore quite incompatible) granularity of two or more agents. An agent communicates a message to some other agent. This activity is done in terms of its own vocabulary (information granules). The other agent has to take advantage of the message being sent to it, interpret (encode) it in its own language and then take further action. This issue requires handling some conceptual underpinnings and quantifying these in terms of some meaningful indexes of communication compatibility. If the vocabulary of the language and/or the granularity of terms used by the two agents are highly incompatible, the communication does not take place.
- Introducing a detailed algorithmic layer using which the agents can realize meaningful communication and quantify the results of such interaction. More precisely, we are interested in the technical insight of a tandem of encoding - decoding mechanisms of granular messages sent between the agents.

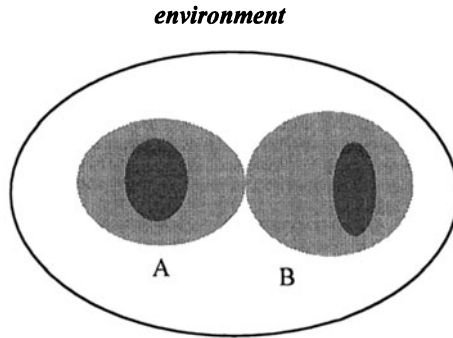


Figure 2. Embedding agents (A and B) in a global communication environment; note a communication interface (communication layer) formed around each agent (a computational architecture of the agent is indicated as a darker core).

The overall architecture of the communication between the agents as schematically portrayed in Figure 2, distinguishes between the numeric core of the agent and its interface layer.

Before moving into technicalities, it is prudent to make some general observations. First, if we are provided with detailed (very specific) information granules, especially those arising at the numeric level, such messages are fully accepted by any agent in spite of the granularity level it operates on. Secondly, we anticipate that the agent may not be capable of handling incoming messages that are too general with regard to

the granularity of its own information granules. There is a broad variety of situations in between that require detailed numeric quantification. To describe the quality of the existing communication processes, let us set up a notation. An input information granule is denoted by X . The collection of the information granules, specific to the agent (A) accepting, is denoted by A_1, A_2, \dots, A_c . We assume that all fuzzy sets in this scenario are normal. The possibility measure $\text{Poss}(X, A_i)$ serves as a measure of interaction (communication) between the agent and the incoming message. The communication index expressing effectiveness of the communication between the agents is defined in the form

$$\text{Comm_index}(X, A_1, A_2, \dots, A_c) = c - \sum_{i=1}^c \langle \text{Poss}(X, A_i) \rangle \quad (1)$$

Note that if X is equal to the entire universe of discourse (yielding $\text{Poss}(X, A_i) = 1$ for all $i = 1, 2, \dots, c$), then the communication index is equal to zero meaning that there is a strong incompatibility in the communication process. Moreover when X tends to be less specific (detailed) then the values of the communication index decrease. The way in which this index decreases depends on the form of X as well as the specific membership functions of A_i 's. Observe that we also average over the moving information granule X of some fixed granularity (this is indicated by $\langle \text{Poss}(X, A_i) \rangle$). The averaging of this type helps us achieve higher relevance of the results. Finally, $\text{Comm_index}(\emptyset, A_1, A_2, \dots, A_c) = 0$. If the collections of the above granules form a fuzzy partition (that is their membership grades sum up to 1 for any element of the universe of discourse), we know that when $\sigma\text{-count}(X) = 1$ then the communication index assumes the value equal to $c-1$. This value can be considered as a reference point when comparing all other situations with the granularity of X being different from 1.

As an example, we consider four Gaussian membership functions of A_i while the incoming message X is treated as an interval of width "2a" distributed around a center "x". The parameters of the Gaussian fuzzy sets are summarized in Table 1; note that we have considered several combinations of the parameters of the membership functions (the same modal values and different spreads). The values of the communication index are plotted in Figure 3. As anticipated, the quality of communication deteriorates when X grows up in size of its granules. This deterioration, however, depends upon the size of the information granules used by the agent to whom the message has been delivered.

The communication index serves as an essential indicator of quality of communication established between the agents. Low values of this index indicate an insufficient level of communication between the agent and the environment or some other agent.

Case	Mean values	Spread (the same for all granules)
1	1, 3, 5, 8	0.5
2	1, 3, 5, 8	2.0

Table 1. Membership functions of the information granules forming a communication interface of the agent: selected collections of the parameters of the Gaussian fuzzy sets

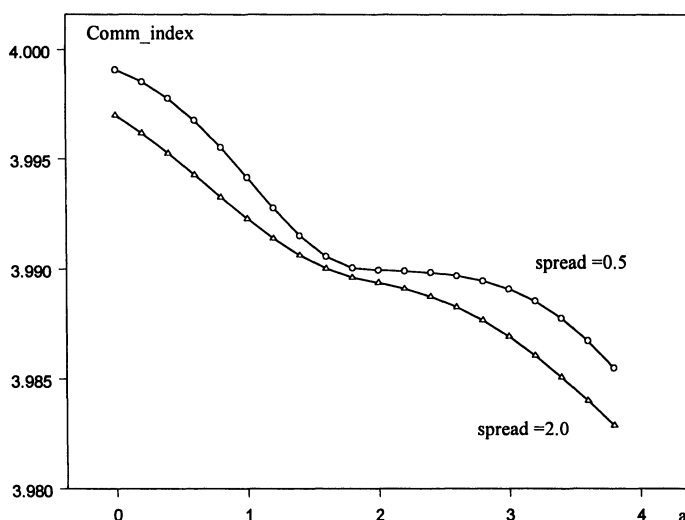


Figure 3. Communication index treated as a function of the size (a) of the set-based information granule.

Let us stress that a necessary condition of effective communication is expressed in the form

$$\sum_{i=1}^c \langle \text{Poss}(X, A_i) \rangle > 0$$

stating that X has to communicate to the agent that falls in its scope of operation (obviously, if all information granules exploited by the agent do not overlap with X , then the communication becomes meaningless).

We can localize agents in a granularity - scope space which is a convenient formalism to describe the essence of communication mechanisms in a general sense, (see Figure 4). By the scope dimension (scope axis) we mean a range of information granules

used by the agents. The granularity coordinate positions the agents as to their specificity and the ability to communicate (where the quality of communication is now articulated in the form of the communication index). For instance, two agents, C and D do not communicate because of their different scope. Agents A and C are close in terms of the granularity of information being used, as well as their scope. As a consequence, it is very likely that they will be communicating quite well.

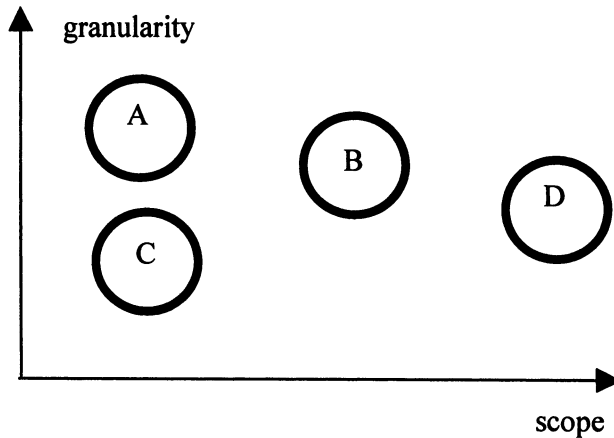


Figure 4. A distribution of agents (A, B, C, and D) in the granularity - scope space.

14. 3 A FUZZY STATE MACHINE AS A GENERIC MODEL OF AN INTELLIGENT AGENT

Finite state machines and automata theory have been a backbone of a variety of models spreading across a spectrum of various disciplines Hill and Peterson (1981), Kandel and Lee (1979), Roth (1995), Schneeweiss (1989), Villa et al. (1997) including hardware systems, industrial controllers, compilers, software requirement analysis, discrete optimization just to name a few. Fuzzy sets have augmented the automata theory in a relatively early phase of their developments; in this regard the reader may refer to the classic monograph authored by Kandel and Lee (1979) where they address various generalizations available in this setting. Let us remind that a finite state machine (automaton) is defined as the structure

$$\langle X, Q, Y, f, g \rangle \quad (4)$$

where X is a set of inputs, Q denotes a set of states, Y is a set of outputs while " f " and " g " are the next-state switching and output functions. The elements of the sets X ,

Q , and Y are binary (0 or 1) or could come from a finite collection of some symbols (a, b, c, \dots) that are afterwards coded in a binary fashion.

The standard way of graphical representation of these machines is through a state graph, that is a graph whose nodes represent states while edges are used to identify transitions between the states. By its nature, at any point of time, the system may reside in only one state (if this property does not hold, we refer to it as a nondeterministic machine). Fuzzy state machines admit a concept of partial belongingness to a given state. In other words, we allow the system to reside in a collection of states. Similarly, we relax the requirement of occurrence of one input symbol, so that several transitions (edges) of the graph are active as well. Figure 5 contrasts finite state machines with their fuzzy set-based extension.

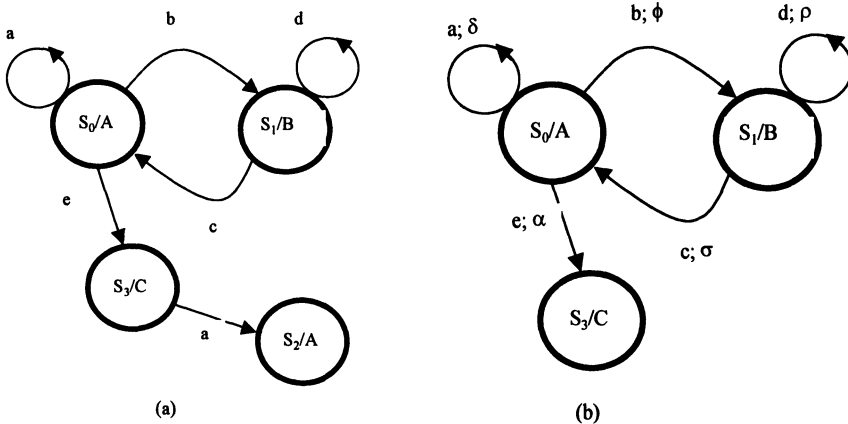


Figure 5. Finite state machines (a) versus fuzzy state machines (b); observe that the latter can have several states and a number of transitions (edges) involved in the dynamics of the machine. $\alpha, \beta, \gamma, \dots$ denote a strength of transition and degree of membership to the state.

In a broader perspective, one has to refer to a number of research pursuits occurring in the area of analysis and use of fuzzy state machines; noteworthy is a diversity of the methods used therein, cf. Diamond et al. (1994), Hirota and Ozawa (1989a-b), Mori et al. (1995), Pedrycz (1995), Bargiela, Pedrycz, 2002; Virant et al. (1995 a-b).

The ability of the fuzzy finite state machines to serve as an efficient model of the agents can be justified in several different ways:

- These automata exhibit a transparent interpretation and show their dynamics in terms of basic logic-inclined entities.
- They can be learned based through an interaction with an environment and other agents. There is a complete suite of the learning schemes and the finite state machines themselves come equipped with a high level of the parametric flexibility residing within the set of adjustable (trainable) parameters (weights) of the logic elements of the machine.
- They can interact with a continuous environment as we admit inputs and outputs that are located in the unit interval.

To cast such fuzzy state machines in the framework of agents, refer to Figure 6. The detailed architecture of the agent and its communication layer are clearly distinguished.

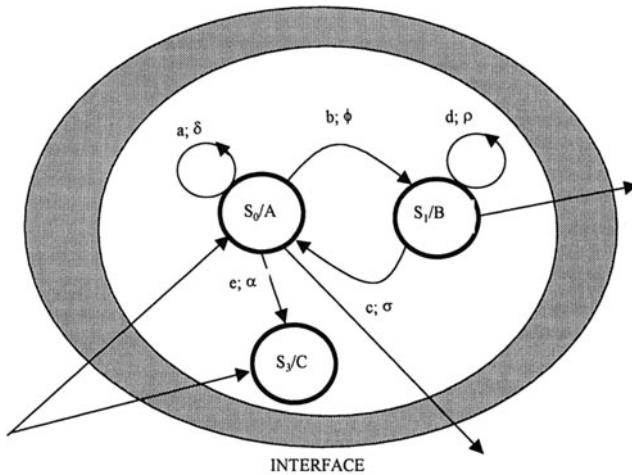


Figure 6. Fuzzy state machines as internal architectures of the agent.

14. 4 THE FUZZY JK FLIP-FLOP AND ITS DYNAMICS

It is instructive to start with a generic definition of the two-valued JK flip-flop in the form of the next-state equation that could be found in any introductory text on digital systems, cf. Hill and Peterson (1981)

$$Q^+ = J\bar{Q} + \bar{K}Q \quad (5)$$

The two inputs (J and K) are used to control the state of the flip-flop. By analyzing the above expression, we can describe the behavior of this system in the tabular form (next-state table) as illustrated below:

Q	JK=00	JK=01	JK=11	JK=10
0	0	0	1	1
1	1	0	0	1

Q^+

Apparently, $J = K = 0$ maintains the same state of the flip-flop, say $Q^+ = Q$. $J=1$ sets the system ($Q^+ = 1$), $K=1$ resets it (that is $Q^+ = 0$) and $J=K=1$ toggles the state (meaning that $Q^+ = \bar{Q}$). The dynamics of the flip-flop can be portrayed in Figure 7. As the variables assume only two values (0-1), the overall pattern of dynamics is quite simple.

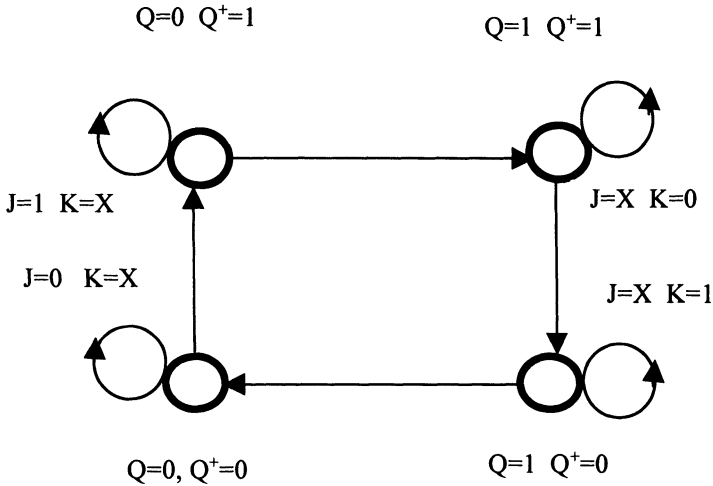


Figure 7. The dynamics of the JK flip-flop showing all transitions between the states $Q-Q^+$

The fundamental generalization of the JK flip-flop has been first proposed in Hirota and Ozawa (1989 a) and then discussed in a series of ensuing papers (Diamond et al, 1994; Pedrycz, 1995; Virant et al, 1999a-b). The formula governing the fuzzy JK flip-flop reads as

$$Q^+ = J\bar{Q} + \bar{K}Q + J\bar{K} \quad (6)$$

Interestingly enough, we can envision it to generalize a group of 1s as they occur in the two-valued Karnaugh map, see Figure 8.

Q	JK=00	JK=01	JK=11	JK=10
0	0	0	1	1
1	1	0	0	1

Figure 8. Karnaugh map showing the three groups of 1s as being used in the expression of the fuzzy JK flip-flop.

As the logic operations are realized in terms of triangular norms, we can rewrite (1) in an explicit manner

$$Q^+ = (Jt\bar{Q})s(\bar{K}tQ)s(Jt\bar{K}) \quad (7)$$

In the sequel, realizing the t-norm as the product operation and the probabilistic sum as the s-norm (i.e. $atb=ab$ and $asb=a+b-ab$), we make (7) more specific, that is

$$Q^+ = J\bar{Q} + \bar{K}Q - J\bar{K}\bar{Q}Q + J\bar{K} - J^2\bar{Q}\bar{K} - J(\bar{K})^2Q + J^2(\bar{K})^2\bar{Q}Q \quad (8)$$

One can easily verify, by a straight inspection, that when confining to J, K, Q in $\{0,1\}$, the fuzzy JK flip-flop subsumes its standard two-valued counterpart.

As one may anticipate, the continuous character of the changes in state and inputs of the flip-flop can result in a very rich pattern of its dynamics. In contrast to the four combinations of the current-next state $Q-Q^+$, we envision an infinite number of possible sequences of states. This effect of continuity of states is illustrated in Figure 9.

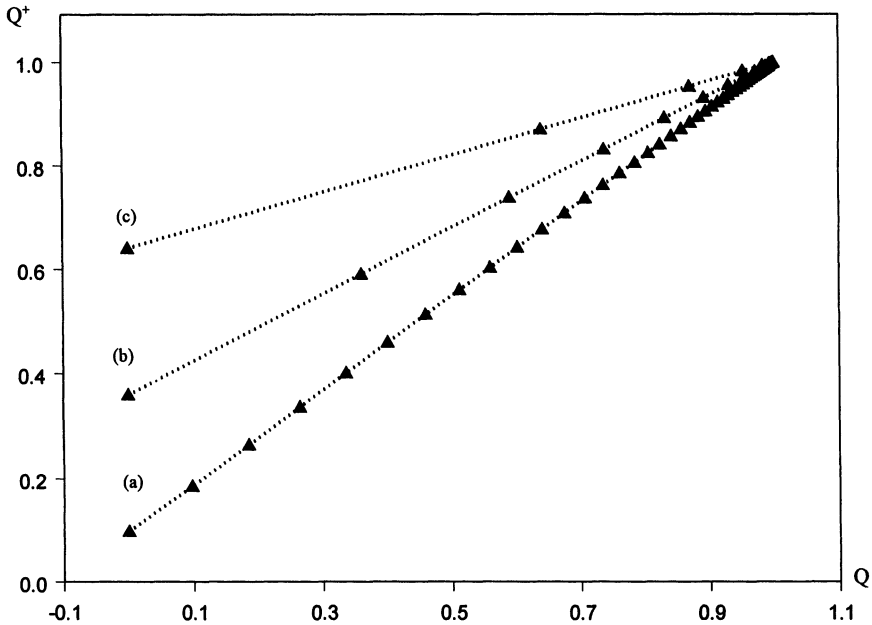


Figure 9. Plots of dynamical patterns of changes observed in the JK flip-flop in the Q - Q^+ space. The starting point is zero, $Q=0$. Moreover $K=0$ while J is equal to 0.05 (a), 0.4 (b), and 0.8 (c) (t-norm: product; s-norm: probabilistic sum).

As seen in Figure 9, we start from the same initial state $Q=0$ and converge to the same final state equal to 1. This is obvious as K is equal to zero and there is a nonzero value of the set input ($J \neq 0$). The speed of changes depends on the value of the set input (J). Note that the reset action is nonexistent (with K equal to zero). The higher the value of J , the faster the changes occurring in the state value (Q^+) of the fuzzy flip-flop. Figure 10 shows the pattern of changes for some configurations of the set and reset inputs. These patterns are more complex and depend which signal prevails. It is essential to notice that depending upon the configuration of the values of the inputs, the system migrates to the values of Q^+ being lower or higher than the initial value of the state.

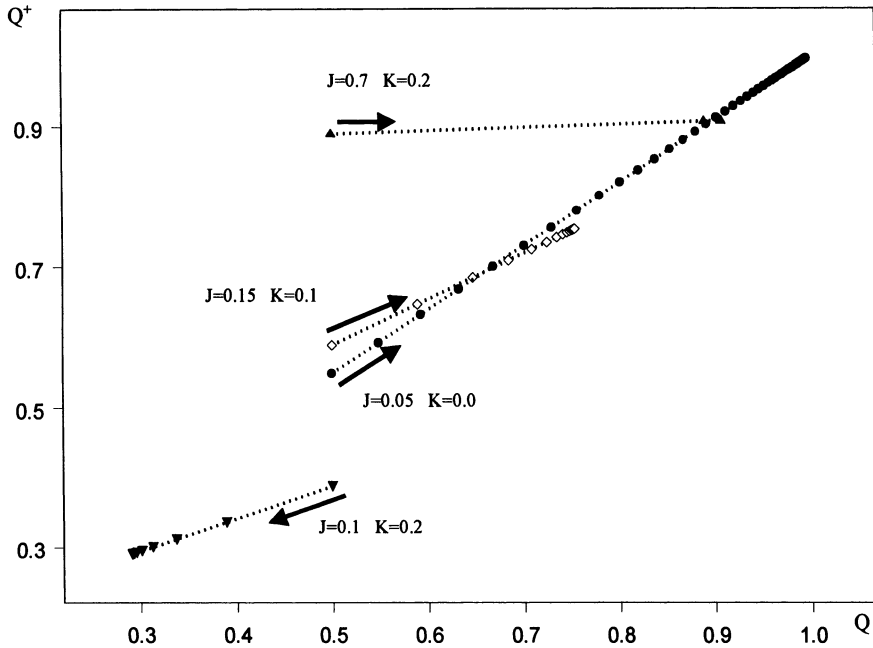


Figure 10. The plot of the dynamics of the JK flip-flop in the Q - Q^+ space. The starting point is 0.5, $Q=0.5$ (t-norm: product; s-norm: probabilistic sum).

14.5 THE DEVELOPMENT OF MOORE TYPE FUZZY STATE MACHINES

In this section, we develop a Moore type of the fuzzy state machine. First, we elaborate on the basic architecture as it generalizes the well-known two-valued counterpart Roth (1995). The Mealy type of fuzzy state machine is developed in the same manner, so this will not be discussed here.

The Architecture

In what follows, we expand the classic Moore type of finite state machine (automaton) to the format of fuzzy inputs and states. Let us recall that the essence of the Moore finite state machine is that the outputs depend on the state but not the input. In other words, the following expressions hold:

$$\begin{aligned} Q^+ &= f(x, Q) \\ y &= g(Q) \end{aligned} \tag{9}$$

where x , y , and Q (Q^+) are vectors of the inputs, outputs, and states. Obviously, in the two-valued case the above are elements of $\{0, 1\}$. It is well-documented (Hill and Peterson, 1981) that the description of the Mealy machine gives rise to the architecture composed of three modules:

- A combinational module (Boolean function) realizing the excitation of J and K inputs of the flip-flops. They indirectly realize the next-state equation (f)
- A combinational module (Boolean function g) mapping the current state (Q) on the outputs of the machine (y)
- A family of JK flip-flops memorizing the state of the system

These three modules along with their interrelationships are portrayed in Figure 11.

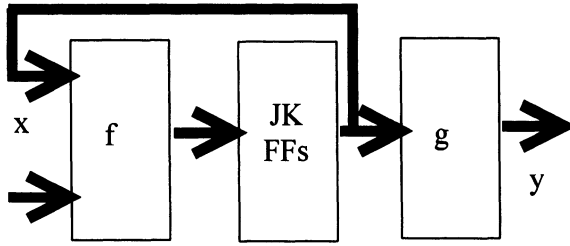


Figure 11. Moore state machine: a general topology composed of two combinational modules (f and g) and a family of JK flip-flops (JK FFs).

From now on, our goal is to generalize it to the format in which it could cope with continuous (fuzzy) variables. This requires a generalization of the basic functional modules such as memory block and two combinational modules. At this point, we can exploit fuzzy JK flip-flops as basic memory elements. The combinational part of the system is realized by so-called logic processors. We discuss them in the next section.

A Logic Processor and its Detailed Topology

In Pedrycz (1995) and Pedrycz and Gomide (1998) proposed were two general classes of fuzzy AND and OR neurons. They serve as generalizations of standard AND and OR digital gates. The AND neuron is a static n -input single output processing element $y = \text{AND}(x; w)$ constructed with the use of fuzzy set operators (t - and s -norms)

$$y = \bigwedge_{i=1}^n (x_i s w_i) \quad (10)$$

Here w denotes a weight vector (connections) of the neuron. In light of the boundary conditions of the triangular norms, we obtain:

- if $w_i = 1$, then the corresponding input has no impact on the output. Moreover, the monotonicity property holds: higher values of w_i 's reduce an impact of x_i on the output of the neuron,
- if w assumes values equal to 0 or 1, then the AND neuron becomes a standard AND gate.

The OR neuron, denoted as $y = \text{OR}(x; w)$, is described in the form:

$$y = \sum_{i=1}^n (x_i t w_i)$$

As before for the AND neuron, the same general properties hold; the boundary conditions are somewhat complementary: the higher the value of the connection, the more evident the impact of the associated input on the output.

The fundamental Shannon's expansion theorem (Peterson and Hill, 1981) states any Boolean function can be represented as a sum of minterms (or equivalently, a product of maxterms). The realization is a two-layer digital network: the first layer has AND gates (realizing required minterms); the second one consists of OR gates that carry out OR-operation on the already constructed minterms.

Fuzzy neurons operate in an environment of continuous variables. An analogy of the Shannon theorem (and the resulting topology of the network) can be realized in the form illustrated in Figure 12. Here AND neurons form a series of generalized minterms. The OR neurons serve as generalized maxterms. This network *approximates* experimental continuous data in a *logic-oriented* manner. In contrast, note that the sum of minterms *represents* Boolean data. The connections of the neurons equip the network with the highly required parametric flexibility. Alluding to the nature of approximation accomplished here, we will be referring to the network as the logic processor (LP).

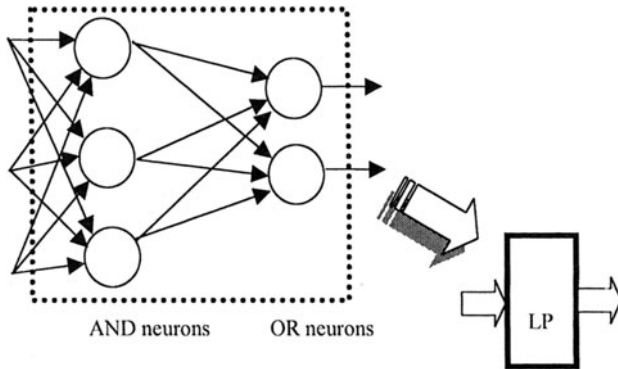


Figure 12. Logic processor (LP): a general topology; the inputs involve both direct and complemented inputs.

A fuzzy Moore State Machine

These logic processors are afterwards used as the building modules in the Moore machine serving as two combinational structures there. The elements of the memory are the fuzzy JK flip-flops.

Now, combining all these functional modules together, we end up with the generalized version of the two-valued Moore machine. Two LPs are used to realize fuzzy combinational functions: the first provides the fuzzy JK flip-flops with the required switching mechanism, the other generates the output function (see Figure 13).

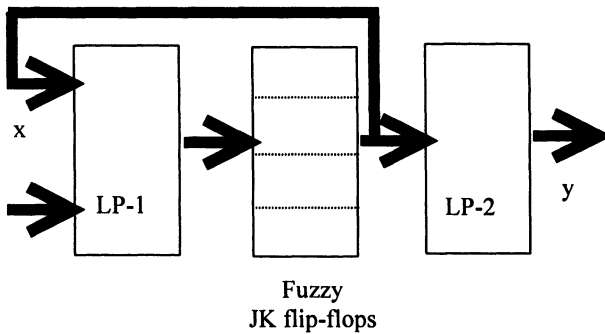


Figure 13. Fuzzy Moore machine: a general architecture.

14. 6 The Learning Scheme

The fuzzy Moore machine comes equipped with the connections of the logic processors that could be easily adjusted. The learning is regarded as a design paradigm of fuzzy sequential systems. We proceed with the design problem formulated as follows

Given an input-output sequence $\{ (x(1), target(1)), (x(2), target(2)), \dots, (x(N), target(N)) \}$ of the agent operating in a certain environment. Design a fuzzy Moore machine that represents (approximates) this sequence. This formulation is somewhat limited as we are concerned with many inputs $(x(k))$ and a single output $(target(k))$, $k=1, 2, \dots, N$. We will consider the multiple input - multiple output case afterwards.

When dealing with the single output, the fuzzy state machine can be arranged in the topology as illustrated in Figure 14.

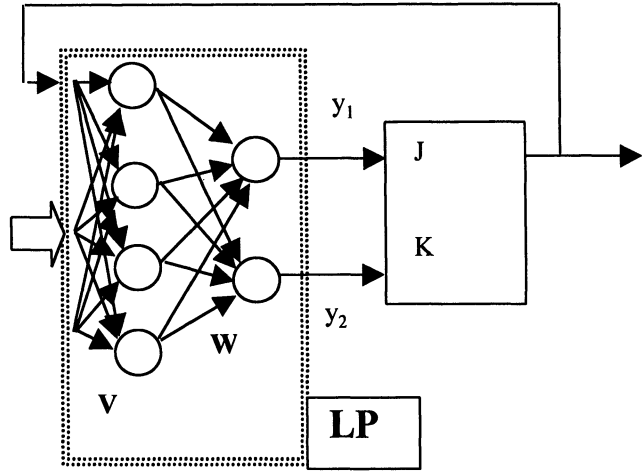


Figure 14. A structure of the fuzzy state machine used in the learning scenario.

Simply, the output function (g) is viewed as an identity and this allows us to eliminate the output logic processor. Only a single fuzzy flip-flop is needed and its state Q is equal to the output, $y = Q$. The remaining logic processor, driving the inputs of the flip-flop, has to be optimized, viz. its connections have to be adjusted so that a certain performance index is minimized. The learning is carried out in the supervised mode. To proceed with the detailed learning scheme, we first fix all necessary notation (see Figure 15).

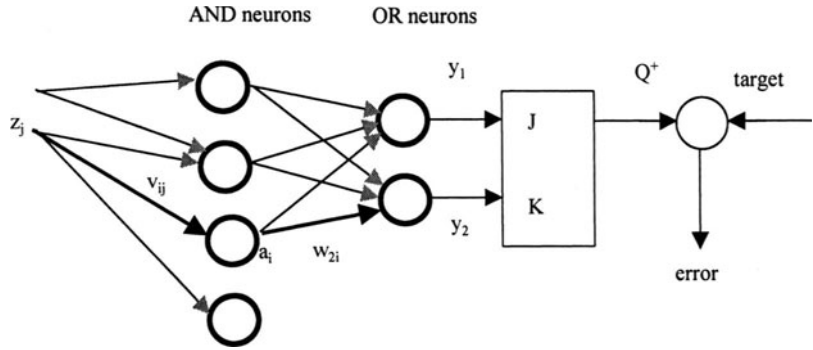


Figure 15. The learning process for fuzzy state machine- a detailed notation.

The logic processor has two outputs, the first driving the set (J) input of the flip-flop and the second used to reset the flip flop. Furthermore the first input is the feedback

from the flip-flop. In total, this gives rise to $m = 2(n+1)$ inputs. The performance index V to be minimized is a standard sum of squared errors:

$$V(\text{conn}) = \sum_{k=1}^{N-1} [\text{target}(k+1) - Q^+(k)]^2 \quad (6)$$

Here **conn** denotes a family of all connections of the LP. More specifically, **conn** is a structure $\{\mathbf{W}, \mathbf{V}\}$ with \mathbf{V} and \mathbf{W} being the arrays of the connections of the AND and OR neurons, respectively, (see Figure 16). Considering (x) , $Q^+(k)$ is governed by (x) and $Q(k) = y(k)$. In computing the performance index (6), we start from $Q^+(1) = y(2)$, $Q(1) = y(1)$, etc. In this case, the objective is to make $\text{target}(2)$ as close to $y(2)$ as possible. Referring to the structure in Fig.16, we have

$$\begin{aligned} J &= y_1, K = y_2 \\ y_1 &= \text{LP}(x(k), \mathbf{W}, \mathbf{V}) \\ y_2 &= \text{LP}(x(k), \mathbf{W}, \mathbf{V}) \end{aligned}$$

The minimization is completed by changing the connections of the logic processor. While the general learning scheme is compact

$$\text{conn}(\text{iter} + 1) = \text{conn}(\text{iter}) - \alpha \nabla_{\text{conn}} V$$

the details can be derived once we have confined ourselves to some specific triangular norms. The derivation is straightforward yet somewhat lengthy. Table 2 summarizes the on-line learning for the product and probabilistic sum being used in the implementation of the fuzzy neurons. As in the on-line learning, the updates of the connections occur after each element of the training set, we drop the index labeling it in the data set. The size of the hidden layer (number of AND neurons) is equal to "h".

$$\text{General update scheme} \quad \text{conn}(\text{iter}+1) = \text{conn}(\text{iter}) - \alpha \nabla_{\text{conn}} V$$

$$\frac{\partial Q}{\partial w_{ij}} = -2(\text{target} - Q^+) \frac{dQ^+}{dw_{ij}}, \quad i=1, 2; \quad j=1, 2, \dots, h$$

$$\frac{dQ^+}{dw_{1j}} = C_1 \frac{dy_1}{dw_{1j}} \quad \text{and} \quad \frac{dQ^+}{dw_{2j}} = C_2 \frac{dy_2}{dw_{2j}}$$

$$C_1 = (1-Q) + (1-K) - 2(1-K)J(1-Q) - Q(1-K)^2$$

$$C_2 = (1-Q)^2 J^2 - 2J(1-K)(Q+J)$$

$$\frac{dy_i}{dw_{ij}} = (1 - A_i) a_i$$

where $A_i = \sum_{\substack{l=1 \\ l \neq j}}^h (w_{il} a_l)$ and "S" stands for the probabilistic sum

$$\frac{\partial Q}{\partial v_{ij}} = -2(\text{target} - Q^+) \frac{dQ^+}{dv_{ij}} \quad i = 1, 2, \dots, h; j = 1, 2, \dots, m$$

$$\frac{dQ^+}{dv_{ij}} = C_1 \frac{dy_1}{dv_{ij}} + C_2 \frac{dy_2}{dv_{ij}}$$

$$\frac{dy_1}{dv_{ij}} = \sum_{i=1}^2 \frac{dy_1}{da_i} \frac{da_i}{dv_{ij}}$$

$$\frac{dy_1}{da_i} = w_{li} (1 - B_i)$$

where

$$B_i = \sum_{\substack{k=1 \\ k \neq i}}^h (w_{ik} a_k)$$

$$\frac{da_i}{dv_{ij}} = C_i (1 - z_j) \text{ and } C_i = \prod_{\substack{t=1 \\ t \neq j}}^m (v_{it} + z_t - v_{it} z_t)$$

Table 2. A detailed on-line learning algorithm for the fuzzy state machine (α -learning rate).

Example 2 Here, we start with a one-dimensional case. The Boolean data set consists of the triples (Q, x, Q^+) :

$$\begin{pmatrix} 000 \\ 011 \\ 101 \\ 110 \end{pmatrix}$$

The learning is realized in the on-line form with the value of α equal to 0.45. The values of the performance index (6) obtained in successive learning epochs are shown in Figure 16.

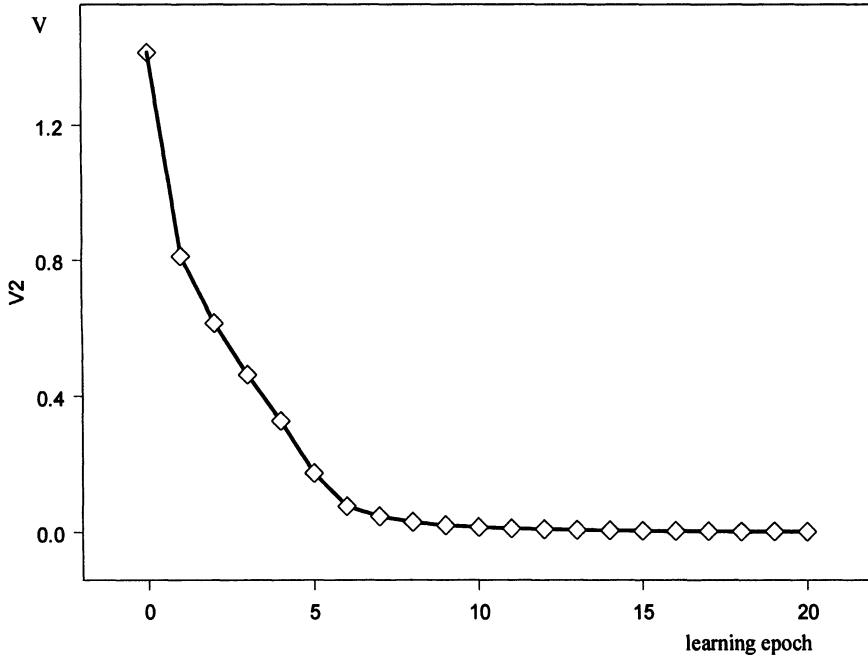


Figure 16. The values of the performance index V in successive learning epochs.

It is evident that the learning is fast and most changes (updates) of the connections of the LP occur at the early phase of the entire process. The optimized connections of the logic processor are summarized below:

- input - hidden layers (the successive columns correspond to the connections originating from the input layer and involving Q , x , and their complements, respectively)

$$V = \begin{pmatrix} 1.000 & 0.000 & 0.998 & 1.000 \\ 0.402 & 0.663 & 0.499 & 1.000 \end{pmatrix}$$

- hidden-output layers (the columns correspond to the outputs of the LP)

$$\mathbf{W} = \begin{pmatrix} 1.000 & 0.000 \\ 1.000 & 0.041 \end{pmatrix}$$

The application of the pruning mechanism (where the most essential connections are retained) leads us to the modified matrices of the connections that could be easily interpreted (the essential connections are indicated)

$$\begin{pmatrix} 1.000 & \mathbf{0.000} & 1.000 & 1.000 \\ \mathbf{0.000} & 1.000 & \mathbf{0.000} & 1.000 \end{pmatrix} \quad \begin{pmatrix} \mathbf{1.000} & 0.000 \\ \mathbf{1.000} & 0.000 \end{pmatrix}$$

Then the formula for the entire LP reads as

$$J=K = x$$

So we ended up with a T flip-flop ($J=K$) (as could have been expected by inspection of the state diagram for the Boolean data, see Figure 17).

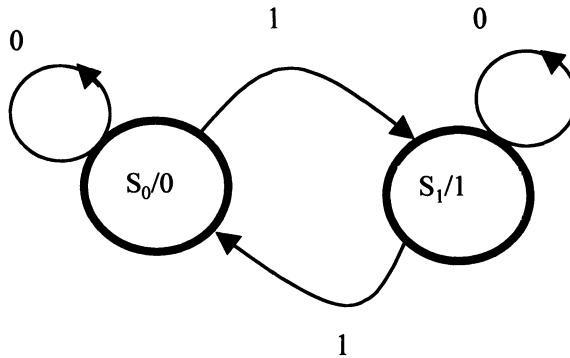


Figure 17. The state diagram of the finite state machine resulting from the Boolean data set; (S/y) denotes state-output pairs.

Moving on with the example, here we are concerned with the 2 input - one input binary data. The data set comes in the format (Q, x_1, x_2, Q^+)

$$\begin{pmatrix} 0000 \\ 0010 \\ 0111 \\ 0101 \\ 1001 \\ 1101 \\ 1110 \\ 1010 \end{pmatrix}$$

The state diagram equivalent to this data set is shown in Figure 18.

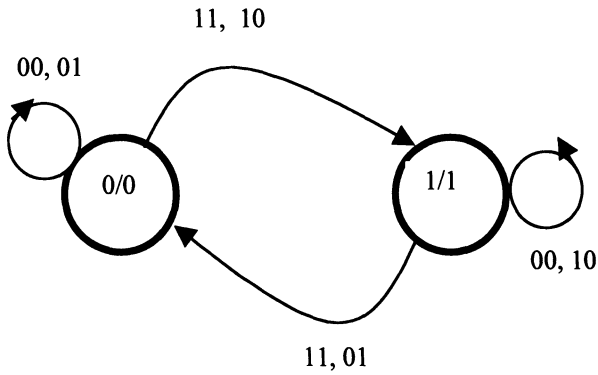


Figure 18. The state diagram corresponding to the binary data set; as we are concerned with the Moore machine the output is equal to the state, $y = Q$.

The values of the minimized performance index obtained in successive learning epochs are shown in Figure 19. We have started from the minimal topology of the LP having $h = 2$ nodes (AND neurons) in the hidden layer. As before, most of the improvements (optimization) take place at the beginning of the entire learning process.

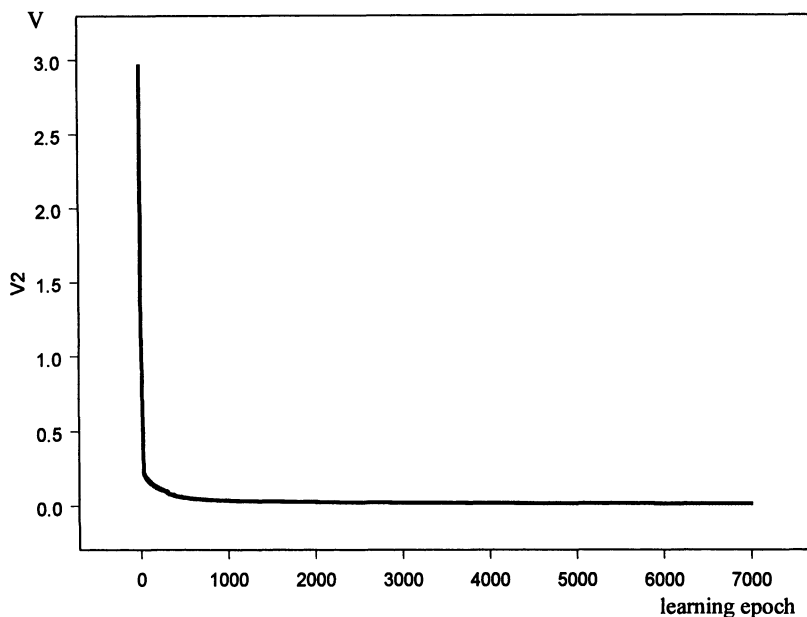


Figure 19. The performance index V as a function of "h" after 7000 learning epochs (learning rate is equal to 0.45).

The value of the performance index is equal 0.0160. At this level, the data are well-represented by the fuzzy state machine. This claim is fully legitimate by looking at the results shown in Table 3. We have slight departures from plain 1s and 0s but this does not prevent us from a clear identification (rounding off) of such.

data	0.000	0.000	1.000	1.000	1.000	1.000	0.000	0.000
FSM	0.000	0.000	1.000	1.000	0.955	0.953	0.047	0.098

Table 3. The results of learning: data and output of the fuzzy state machine (FSM)

As before, the connections are organized in the matrix form. For the input - hidden layer, the order is: Q , x_1 , x_2 followed by their complements (column-wise). The rows correspond to the two nodes in the hidden layer:

$$V = \begin{pmatrix} 1.000 & 0.000 & 1.000 & 0.000 & 1.000 & 1.000 \\ 0.501 & 0.947 & 0.0496 & 1.000 & 1.000 & 1.000 \end{pmatrix}$$

The connections for the hidden-output layer are summarized below

$$\mathbf{W} = \begin{pmatrix} 1.000 & 0.000 \\ 1.000 & 0.953 \end{pmatrix}$$

The connections are in $[0,1]$. The original data are binary. It is therefore of interest to prune (or better say, binarize) the LP to get a clear understanding as to the Boolean functions realized by the system. Following the pruning formula, we get the connections (additionally the essential connections that contribute to the overall interpretation of the network are put in boldface)

$$\mathbf{V} = \begin{pmatrix} 1.000 & \mathbf{0.000} & 1.000 & \mathbf{0.000} & 1.000 & 1.000 \\ 1.000 & 1.000 & \mathbf{0.000} & 1.000 & 1.000 & 1.000 \end{pmatrix} \quad \mathbf{W} = \begin{pmatrix} \mathbf{1.000} & 0.000 \\ \mathbf{1.000} & \mathbf{1.000} \end{pmatrix}$$

In the sequel, they help us to describe the functions:

- for the AND neurons in the hidden layer,

$$\text{hidden}_1 = x_1 \overline{Q}$$

$$\text{hidden}_2 = x_2$$

- for the OR neurons in the output layer,

$$J = \text{hidden}_1 = x_1 \overline{Q}$$

$$K = \text{hidden}_1 + \text{hidden}_2 = x_1 \overline{Q} + x_2$$

How does these expression compare to the result of the "standard" design of such digital system? We can repeat the design process following excitation tables for the JK flip-flop [6][14]. This leads us to the Karnaugh maps for the J and K input of the flip-flop (see Figure 20).

$Q \setminus x_1 x_2$	00	01	11	10
0	0	0	1	1
1	X	0	0	X

J

$Q \setminus x_1 x_2$	00	01	11	10
0	X	X	X	X
1	X	1	1	X

K

Figure 20. K-maps for J and K inputs of the flip-flop.

By encircling the neighboring 1s (and taking advantage of the existing don't care conditions - X), we derive the expressions

$$J = x_1 \overline{Q}$$

$$K = x_2$$

The Boolean expression for the J input is the same as the one obtained through learning. The expression for the K input is different: simply what has been learned includes one extra term (that is composed of don't care conditions only). In other words, this portion of the formula is not excessive. Its inclusion, though, does not make any harm to the expression and does not cause any extra hardware as this portion has been already used to implement the set input of the flip-flop, $K = J + x_2$.

To summarize these two experiments: the fuzzy state machine (agent) has been *learned* rather than being designed -- it comes as an interesting endeavor of some practical implications (especially in light of the current tendency of evolving and learning hardware as we have already witnessed in evolvable hardware).

The experiment was carried out for the increasing size of the hidden layer by changing the number of the AND neurons; the results are summarized below:

h	3	4	5
Performance index V	0.0457	0.0148	0.0111

Evidently, the increase of the size of the hidden layer does not substantially reduce the values of the performance index. In this case, it could be attributed to the relatively simple problem we are dealing with.

14. 7 CONCLUSIONS

Granular worlds come with their underlying computing facilities and these were the objective of this chapter. We have studied a concept of agents as being embedded in the framework of granular computing. Two main issues have been studied in depth: the communication mechanism between agents and the detailed architecture of the agent itself. The quality of communication is quantified with the aid of the communication index. In relation to the first issue, each agent being mapped in the granularity - scope space can be characterized in sense of its communication abilities with other agents and/or external environment. The language of granular computing employed in this study concerns fuzzy sets. The subsequent communication issues are expressed in the same language (namely we assumed that all agents are embedded in the same conceptual environment). The choice of this granular environment was implied by the progress in the area of fuzzy automata (Kandel and Lee, 1979; Hirota

and Ozawa, 1998; Pedrycz,). It is worth stating that as the automata theory arises at a relatively high level of abstraction, the overall development is general enough to be applied in other formal environments of granular information. We have showed that the design of the agent itself (whose underlying topology is reflected in the form of the fuzzy state machine) can be carried out through learning that gives rise to their autonomous behavior.

REFERENCES

- Bargiela, A., Pedrycz, W., Hirota, K. (2002), Logic-based granular prototyping, *Soft Computing and Intelligent Systems Conference, SCIS 2002*, Tsukuba, Japan, Oct. 2002.
- Cicalese, F., Di Nola, A., Loia, V. (1999), A fuzzy evolutionary framework for adaptive agents, *Proc 13th Int ACM symp. of Applied Computing*, 29 Feb - 2 March, 1999, San Antonio, TX, ACM Press.
- Cerri S.A. (1997), Shifting the focus from control to communication: the STReams Objects Environments model of communicating agents. In: *Lecture Notes in Artificial Intelligence*, vol. 1624, Springer-Verlag, Berlin, pp. 115-131.
- Diamond, J., Pedrycz, W., McLeod, R.D.(1994), Fuzzy J-K flip-flop as computational structures: design and implementation, *IEEE Trans. on Circuits Systems II*, **41**, 215-226.
- Hill, F.J., Peterson, G.R.(1981), *Introduction to Switching Theory & Logic Design*, J. Wiley, N. York.
- Hirota, K., Ozawa, N.(1989a), The concept of fuzzy flip-flop, *IEEE Trans. on Systems, Man, and Cybernetics*, **19**, 980-997.
- Hirota, K. Ozawa, N. (1989b), Fuzzy flip-flop and fuzzy registers, *Fuzzy Sets and Systems*, **32**, 139-148.
- Kafura, D., Briot, J.P. (1998), Actors and agents, *IEEE Concurrency*, **6**, 24-29.
- Kandel, A., Lee, S.C. (1979), *Fuzzy Switching and Automata. Theory and Applications*, Crane, Russak, New York.
- Mori, Y., Otsuka, K., Mukaidono, M. (1995), Properties of fuzzy sequential circuit using fuzzy transition matrix and their design method, In: *Proc. IEEE Conf.*, Yokohama, Japan, pp. 2133-2138.
- Osman, T., Bargiela, A. (2000), FADI: A fault tolerant environment for open distributed computing, *IEE Proceedings Software*, **147**(3), 91-99.
- Pedrycz, W. (1995), *Fuzzy Set Engineering*, CRC Press, Boca Raton, FL.
- Pedrycz, W., F. Gomide, F. (1998), *Fuzzy Sets: An Introduction. Analysis and Design*, MIT Press.
- Roth, C.H. (1995), *Fundamentals of Logic Design*, PWS Publishing Company, Boston.

Schneeweiss, W.G., (1989), *Boolean Functions with Engineering Applications and Computer Programs*, Springer-Verlag, Berlin.

Villa, T., Kam, T., Brayton, R.K., Sangiovanni-Vincentelli, A. (1997), *Synthesis of Finite State Machines*, Kluwer Academic Publishers, Boston.

Virant, J., Zimic, N., Mraz, M. (1999a), Fuzzy sequential circuits and automata, In: C.T. Leondes (ed.), *Fuzzy Theory Systems*, vol. IV, Academic Press, San Diego, pp.1599-1653.

Virant, J., Zimic, N., Mraz, M. (1999b), T-type fuzzy memory cells, *Fuzzy Sets and Systems*, **102**, 175-183.

Wagealla, W., Osman, T., Bargiela, A. (2002), Error detection algorithm for agent-based distributed applications, Agent Based Simulations Conference, Passau, 106-110.

PART IV

GRANULAR SYSTEMS APPLICATIONS



SELF-ORGANIZING MAPS IN THE DESIGN AND PROCESSING OF GRANULAR INFORMATION

15. 1 INTRODUCTION

In this chapter, we concentrate on a granular data analysis, especially studying ways of information granulation. We show how information granules are constructed by a designer/user via a visual inspection of self-organizing maps (SOMs). SOMs are commonly used neural network architectures realizing a paradigm of unsupervised learning. The crux of the approach proposed here lies in the following

- a high level of interaction with user – it is worth stressing that the constructs (information granules) are delineated by a human on a basis of visualization of highly dimensional data,
- a solid support of the development of information granules cast in the framework of sets and fuzzy sets.

We elaborate on how self-organizing maps create a user-friendly and interactive visualization tool that helps user/software designer inspect various alternatives and develop a thorough insight into the structure of the visually formed information granules. The experimental part includes both synthetic as well as real-world data. The latter consists of two standard machine learning data sets (available at UC at Irvine). There are also two case studies dealing with software engineering data (software quality) and ECG signals. In relation to these data sets, we show how using self-organizing maps we can grow clusters in a dynamic fashion thus explicitly capture relationships between the software measures and quantify these dependencies for larger and less homogeneous clusters of software modules or groups ECG signals.

15. 2 SELF-ORGANIZING MAPS

The concept of a self-organizing map (SOM) has been originally coined by Kohonen (1982, 1995). There has been a wealth of research in this area that addresses a diversity of applications and further conceptual and algorithmic enhancements of the

original idea, cf. Kohonen et al. (1996), Oja and Kaski (1999). As usually emphasized in the literature, SOMs are regarded as regular neural structures (neural networks) composed of a grid of artificial neurons that attempt to visualize highly dimensional data in a low-dimensional structure, usually emerging in the form of a two- or three-dimensional map. To make such visualization meaningful, an ultimate requirement is that such low-dimensional representation of the originally high-dimensional data has to preserve *topological* properties of the data set. In a nutshell, this means that two data points (patterns) that are close each other in the original feature space should retain this similarity (or closeness) when it comes to their representation (mapping) in the reduced, low-dimensional space in which they are visualized. And, reciprocally: two distant patterns in the original feature space should retain their distant location in the low-dimensional space. Being more descriptive, SOM performs as a *computer eye* that helps us gain insight into the structure of the data set and observe relationships occurring between the patterns being originally located in a highly dimensional space. In this way, we can confine ourselves to the two dimensional map that apparently helps us to witness all essential relationships between the data as well as dependencies between the software measures themselves. In spite of the existing variations, the generic SOM architecture (as well as the learning algorithm) remains basically the same. Below we summarize the essence of underlying self-organization algorithm that realizes a certain form of unsupervised learning.

Before proceeding with the detailed computations, we introduce all necessary notation. " n " software measures are organized in a vector X of real numbers situated in the n -dimensional space of real numbers, R^n . The SOM is a collection of linear neurons organized in the form of a regular two-dimensional grid (array), Figure 1.

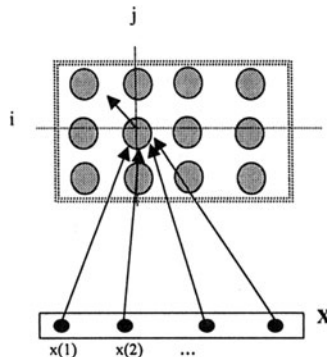


Figure 1. A basic topology of the self-organizing map constructed as a grid of identical processing units (neurons).

In general, such a grid of neurons grid may consist of " p " rows and " r " columns; quite commonly we confine ourselves to the square array of " p " \times " p " elements

(neurons). Each neuron is equipped with modifiable connections $w(i,j)$ and computes a distance function between its connections and the corresponding input x

$$y(i,j) = d(w(i,j), x) \quad (1)$$

where the pair (i,j) denotes a certain (i,j) position of the neuron in the array. x is an input to all neurons and $d(.,.)$ denotes a distance between the connections and the input. The same input x affects all neurons. The neuron with the shortest distance between the input and the connections becomes activated to the highest extent and is called winning neuron. Let us denote its coordinates by (i_0, j_0) . More precisely, we have

$$(i_0, j_0) = \arg \min_{(i,j)} d(w(i,j), x) \quad (2)$$

The winning neuron matches (responds to) x . As a winner of this competition, we reward the neuron and allow it to modify the connections so that they are even closer to the input data. The update mechanism is governed by the expression

$$w_new(i_0, j_0) = w(i_0, j_0) + \alpha(x - w(i_0, j_0)) \quad (3)$$

where α denotes a learning rate, $\alpha > 0$. The higher the learning rate, the more vigorous updates of the connections. In addition to the changes of the connections of the winning node (neuron), we allow this neuron to affect its neighbors. The way in which this influence is quantified is expressed via a neighbor function $\Phi(i, j, i_0, j_0)$. In general, this function satisfies two intuitively appealing conditions: (a) it attains maximum equal to one for the winning node, $i = i_0, j = j_0$ and (b) when the node is apart from the winning node, the value of the function gets lower (in other words, the updates are less vigorous). Evidently, there are also nodes where the neighbor function zeroes. Considering the above, we rewrite (1) in the following form

$$w_new(i, j) = w(i_0, j_0) + \alpha \Phi(i, j, i_0, j_0)(x - w(i, j)) \quad (4)$$

In the ensuing experiments, we use the neighbor function in the form

$$\Phi(i,j,i_0,j_0) = \exp(-\beta((i-i_0)^2 + (j-j_0)^2))$$

with the parameter β (equal to 0.1 or 0.05 depending upon the series of experiments). The role of this parameter is to model the spread of the neighborhood.

The above update expression (4) applies to all the nodes (i, j) of the map. As we iterate (update) the connections, the neighbor function shrinks: at the beginning of updates we start with a large region of updates and when the learning settles down,

we start reducing the size of the neighborhood. For instance, one may think of a linear decrease of its size.

The number of iterations is either specified in advance or the learning terminates once there are no significant changes in the connections of the neurons.

The distance $d(\mathbf{x}, \mathbf{w})$ can be defined in many different ways. A general class worth considering here is that of the Minkowski distance. As a matter of fact, this class of distances constitutes the most general class of distance measure. Consider two vectors of real numbers, \mathbf{a} and \mathbf{b} defined in \mathbf{R}^n . The distance $d_M(\mathbf{a}, \mathbf{b})$ is defined as

$$d_M(\mathbf{a}, \mathbf{b}) = \left(\sum_{i=1}^n |a_i - b_i|^p \right)^{1/p}$$

where "p" is a coefficient assuming values greater or equal to 1.

Interestingly, there are several examples of the Minkowski distance, including the Hamming distance, usually referred to as a city-block distance, the Euclidean distance, and the Tschetschev distance. They are special cases of the Minkowski distance for $p=1$, 2, and infinity. We have several general options

- Hamming distance $d_H(\mathbf{a}, \mathbf{b}) = \left(\sum_{i=1}^n |a_i - b_i| \right)$
- Euclidean distance $d_E(\mathbf{a}, \mathbf{b}) = \left(\sum_{i=1}^n |a_i - b_i|^2 \right)^{1/2}$
- Tschetschev distance $d_T(\mathbf{a}, \mathbf{b}) = \max_{i=1,2,\dots,n} |a_i - b_i|$

The Euclidean distance is the most commonly used. The Hamming distance dwells on the absolute differences between the coordinates and promotes robustness of the resulting constructs (viz. the map) meaning that some slight changes to the data should not affect the configuration (arrangement of the groups) as being more "stable" and general. The Tschetschev distance takes into consideration the maximal distance over the coordinates of \mathbf{a} and \mathbf{b} .

Dealing with raw measures poses the risk that one software measure may become predominant, simply because its domain includes larger numbers (that is the range of the measure is high). Therefore, the distance function is computed for normalized rather than raw data. In the sequel, the SOM exploits these transformed software measures. Two common ways of normalization are usually pursued, the linear and

statistical normalization. In the linear normalization, the original variable is normalized to the unit interval [0,1] via a simple linear transformation

$$x_{\text{normalized}} = \frac{x_{\text{original}} - x_{\min}}{x_{\max} - x_{\min}}$$

where x_{\min} and x_{\max} are the minimal and maximal value of the variable encountered in the data. The statistical normalization uses the mean \bar{x} and the standard deviation σ_x of the variable

$$x_{\text{normalized}} = \frac{x_{\text{original}} - \bar{x}}{\sigma_x}$$

The logistic normalization uses a standard sigmoid function where the normalized version of " x_{original} " reads as

$$x_{\text{normalized}} = \frac{1}{1 + \exp(-x_{\text{original}})}$$

In addition, when observing the activity of the individual neurons in the grid, some of them may be excessively "active" and winning most the time. The other neurons tend to become "idle". This uneven activity pattern is undesired and should be avoided. In order to promote more even activity across the network, we make the learning frequency - sensitive by penalizing the frequently winning nodes and increasing the distance function between the patterns (inputs) and the connections of the winning node. For instance, instead of the original distance $d(x, w)$, we use the expression $(1+\epsilon)d(x, w)$ where ϵ is a positive constant modeling the effect of intentionally increased distance between x and w . The higher the value of ϵ , the more substantial the increase in the effective distance between the pattern and the neuron.

At this point, it is worth mentioning here that FCM builds an explicit list of clusters in the form of so-called partition matrix. It is instructive to contrast the SOM with another class of clustering methods driven by a certain objective function such as C-MEANS and its variations. SOM and FCM are complementary and so are their advantages and shortcoming. FCM requires the number of groups (clusters) to be defined in advance. It is guided by a certain performance index (objective function) and the solution comes in a clear form of a certain partition matrix. In contrast, SOM is more user-oriented. There is no number of clusters (group) that needs to be specified in advance. This may be regarded as an evident advantage (as usually we may not like to commit ourselves to the identification of the number of clusters -- as a matter of fact, during any initial phase of data analysis one may not have a clearly

defined opinion as to this parameter). Obviously, the advantage may convert into a drawback when it comes to the format in which the final results are presented. In the basic form of the SOM, there are no provisions to delineate the clusters automatically and a human intervention may be required. All in all, this could not be that limiting as the visualization of data in the SOM may be exercised to a high degree.

Revealing Structure in Data by Cluster Growing

SOMs, in contrast to other methods of unsupervised learning (such as FCM, ISODATA and alike) do not require an explicitly defined number of clusters. The identification of the clusters is left to the user. In Kohonen et al. (1996) proposed was a method of delineating the boundaries of the clusters (groups) as they emerge on the map. Considering a certain pair of coordinates (i,j) , we calculate the changes of the connections of the neurons located in the neighborhood of this location on the map,

$$\partial W(i, j) = \text{Median}(w_{ij} - w_{\Omega_{ij}}) \quad (5)$$

Where Ω_{ij} is a neighborhood of the (i,j) th node of the SOM and $w_{\Omega_{ij}}$ denotes the connection of the neuron belonging to this neighborhood; refer to Figure 2. (Obviously, depending upon the location on the map, we have to properly handle the boundary conditions in the calculations of the above expression).

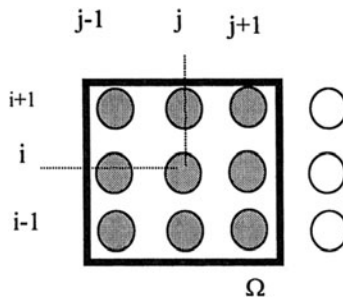


Figure 2. Computing the cumulative changes to the connections at (i,j) coordinate.

If there are no significant changes to the connections in this region, the values of the above expression are low. This naturally makes these entries of the map homogeneous suggesting that such elements of the map form a contiguous region (viz. cluster). Otherwise we may envision this entry (i,j) to form a part of the boundary between the clusters.

Quite naturally, we may expect a high homogeneity of data falling inside a given cluster. By growing dynamically clusters in a way discussed above, we can also monitor and describe the characteristics of the clusters in the statistical form, say by determining a correlation matrix for the corresponding features.

Overall, the developed SOM is fully characterized by a matrix of connections of its neurons, that is $\mathbf{W} = [w(i,j)]$, $i=1,2, \dots, p$, $j=1, 2, \dots, p$ (note that we are now dealing with a squared p by p grid of the neurons). The simplest visualization scenario one can envision is to map the original data on the map so in this manner we get a certain insight into the structure of the data in a highly-dimensional space. For instance, we can state that $x(1)$ and $x(6)$ are similar because they “activate” two neighboring neurons on the map. A visualization of the relative position of the patterns is a main advantage of the SOM. Moreover, by a careful arrangement of the weight matrix into several planes (arrays) we can produce a variety of important views at the data. We introduce such concepts as a weight, region(cluster) and data density map.

15.3 ASSOCIATED SELF-ORGANIZING MAPS

The associated maps come as a result of a different organization of \mathbf{W} or some slight modifications of the original connections. The intent of this arrangement is to come up with a better visualization of the data and relationships existing between the variables (features).

Weight Maps

Obviously the weight matrix \mathbf{W} can be viewed as a pile of layers of p by p maps indexed by the variables, see Figure 3. That is we regard \mathbf{W} as a collection of two-dimensional matrices each corresponding to a certain feature of the pattern, say

$$[w_1(i,j)] \quad [w_2(i,j)] \quad \dots [w_n(i,j)]$$

Each of these matrices contains information about the weights (or the features of the patterns as the weights tend to follow the features once the self-organization has been completed).

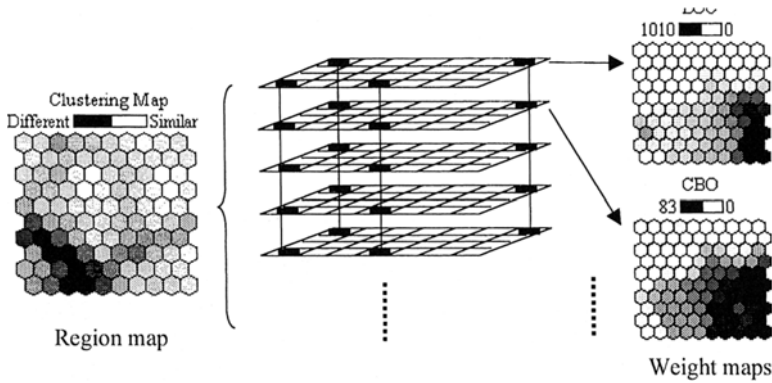


Figure 3. A concept of associated SOM maps: region and weight maps.

The most useful information we can get from these weight maps deals with an identification of possible associations between the features. If the two weight maps are very similar, this implies that the two corresponding features they represent are highly related. If two weight maps are very dissimilar, this means the two variables they represent are not closely interrelated. In addition, we can also determine the feature association for a data subset. For example, two weight maps can be very similar in the upper-right corner but are very dissimilar in other area of the map. This means only the data located in the upper-right corner are highly related. One should note that the identification of relationships is carried out in a visual mode and we do not allude to any measure of association such as a correlation coefficient. Nevertheless, this aspect is highly supportive in a descriptive data analysis and helps the designer understand the essence of the relationships between the features. In the sequel, it may lead to the identification of possible redundancies of some features (e.g., we state that two features are redundant if their weight maps are very close each other).

Region (clustering) Map

A slight transformation (summarization) of the original map \mathbf{W} allows us to visualize homogeneous regions in the map, viz. the regions in which the data are very similar. Furthermore we should be able to form boundaries between such homogeneous regions of the map. Owing to the character of this transformation, we will be referring to the resulting areas as clusters and calling the map a region (or clustering) map. The calculations leading to the region map are straightforward: for each location of the map, say (i,j) we compute distances between the weight vectors of its closest neighbors, like $(i-1, j)$, $(i, j-1)$, $(i, j+1)$..., etc. that is

$$d(\mathbf{w}(i,j), \mathbf{w}(i-1,j))$$

$$d(\mathbf{w}(i,j), \mathbf{w}(i,j+1))$$

....

and take a median of these differences, that is $\text{med}_{\mathcal{I}}[\Delta(i,j)]$ with \mathcal{I} treated as a neighborhood of this particular location of the map. The neighborhood \mathcal{I} functions in a same manner as commonly encountered in image processing (not surprising, the SOM is a digital image). The neighborhood consists of 8 cells of SOM (pixels) surrounding the given neuron of the map. This median is regarded this as a measure of homogeneity of the nearest neighbor of the (i,j) location of the map. At the visual end, we map the median on a certain level of brightness to each of these results that gives us a useful vehicle of identifying regions in the map that are highly homogeneous. Likewise, the entries with dark color form a boundary between the homogeneous regions. Clusters can be easily identified by finding areas of higher level of brightness being surrounded by these dark boundaries. For some data set, there are distinct clusters, so in the clustering map, the dark boundaries are clearly visible. There could be cases where data are inherently scattered, so in the clustering map, we may not see clear dark boundaries. In Figure 3, the region (clustering) map reveals two clusters: the one quite extensive that covers an upper part of the map and the smaller one. The clustering map is an important vehicle for a visual inspection of the structure in the data. It delivers a strong support for descriptive modeling: the designer can easily understand how structure looks like in terms of clusters. In particular, one can analyze the size of the clusters, their location in the map (that tells about closeness and possible linkages between the clusters). By looking at the boundaries between the clusters, the region map tells us how strongly these clusters are identified as separate entities distinct from each other. Overall, we can look at the region map as a granular signature of the data. These visualization aspects of SOMs underline their character as a user-friendly vehicle of descriptive data analysis. In this context, it also points out at the essential differences between SOMs and other popular clustering techniques driven by objective function minimization (say FCM and alike). Note that while FCM solves an interesting and well-defined optimization problem but does not provide with the same interactive environment for data analysis.

It is worth stressing that the homogeneous regions of the SOM could be detected in an automatic manner (as discussed in Oja and Kaski (1999)). While attractive per se, the formation of the regions is affected by the values of some parameters (quite often difficult to adjust) that are not transparent to the user. The position promoted in this study is that the user/designer should play a dominant role in the determination of the regions in the map.

Data Distribution Map

The previous maps were formed directly from the general map \mathbf{W} produced through self-organization. It is advantageous to supplement all these maps with a data

distribution (density) map. This map shows (again on a certain brightness scale) a distribution of data as they are allocated to the individual neurons on the map.

Following the assumed visualization scheme, the darker the color of the neuron, the more patterns invoked the neuron as the winning one, see Figure 4.

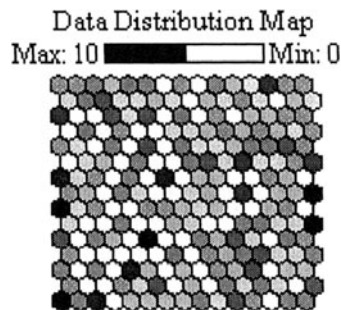


Figure 4. An example data density map; the darker neurons identify groups of higher data density. These regions should be analyzed in conjunction with the region (clustering) map.

The data density map can be used in conjunction with the region map as it helps us indicate how much patterns are behind the given cluster. In this sense, we may eventually abandon a certain cluster in our descriptive analysis as not carrying enough experimental evidence. Moreover, the data density map helps reveal some evident learning problem related to a few frequently winning nodes (neurons); to alleviate this discrepancy, we can introduce a frequency sensitivity component in the learning process. Overall, the sequence realized so far can be described in the following manner

highly dimensional data \Rightarrow SOM \Rightarrow regions, data density maps, feature maps \Rightarrow interactive user-driven descriptive analysis

15. 4 EXPERIMENTS – SYNTHETIC AND MACHINE LEARNING DATA

We take a relatively simple low-dimensional example in which we know the structure in the data set and observe what structure is revealed by the self-organizing map. A four-dimensional synthetic data set is generated by a uniform random generator (random ()) where each group is shifted as follows

$$x_1 = \text{random}() + m_1, x_2 = \text{random}() + m_2, x_3 = \text{random}() + m_3, x_4 = \text{random}() + m_4$$

Four groups were generated, each of them consisting of 2,000 data points. The shift parameters describing each group are listed below

Group no.	m_1	m_2	m_3	m_4
1	0.1	0.5	0.9	0.0
2	0.9	0.1	0.0	0.4
3	0.5	0.9	0.8	0.5
4	0.2	0.4	0.2	1.0

We experimented with two sizes of the map; in the first case it has a 10 by 10 grid of neurons, in the second structure the grid was increased to the size 15 by 15. The learning was carried out for 1,000 epochs (this parameters was chosen on an experimental basis; the optimization was terminated once no significant changes to the connections have been observed). The experimental finding was that after that no substantial changes in the connections of the neurons (and equivalently the structure of the data revealed by the map) have been observed. After the training, the structure of the data was revealed quite profoundly as illustrated in Figure 5 and 6.

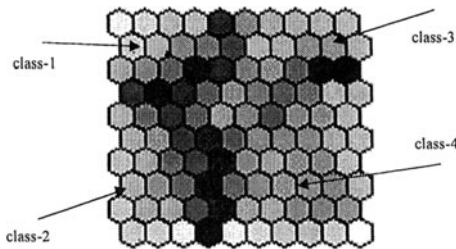


Figure 5(a). The structure revealed by the self-organizing map after the training; neurons marked in darker color delineate regions of high homogeneity both for the 10 by 10 map.

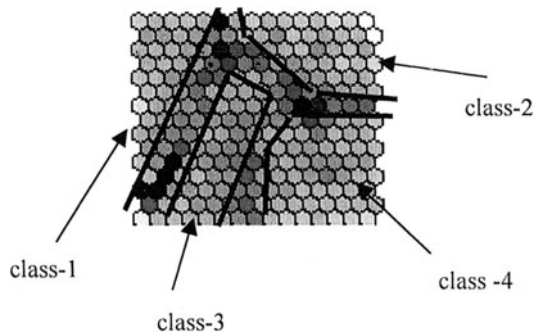


Figure 5(b). The structure revealed by the self-organizing map after the training; neurons marked in darker color delineate regions of high homogeneity both for the 15 by 15 map.

The differences are even more profound and the clusters are clearly delineated when the size of the map was increased to 25 neurons per column/row.

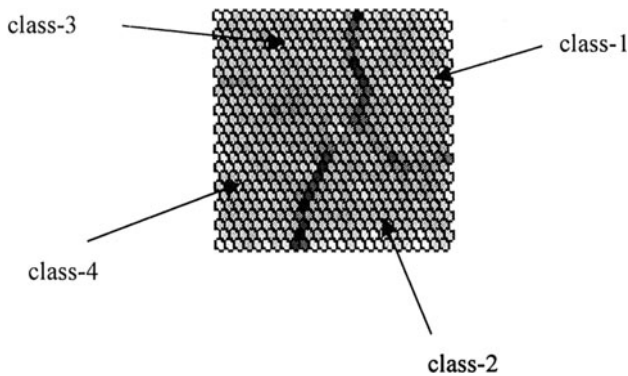


Figure 6. Homogeneous regions in the 25 by 25 SOM.

The distribution map, Figure 7, provides us with the qualitatively the same picture as before yet now it comes with more details. Note that the boundaries are formed by a few data points that are different from the rest of the patterns

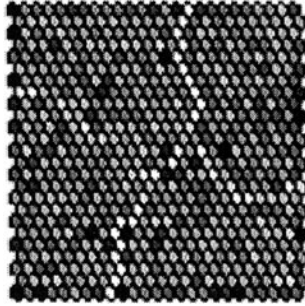


Figure 7. The density distribution of data (associated self-organizing map); the darkest entries correspond to 33 patterns allocated to the respective neuron, the brightest points allocate zero patterns. Note that the boundaries exhibit a very low density of data points.

Finally, the distribution of the features on the map is shown in Figure 8. This provides us with another option to investigate relationships between the features (variables). By visual inspection, we immediately learn that feature 2 and 3 are more “related” (in a visual sense) than the first and second feature. On the other hand, there is a relationship between the first and fourth feature that occurs only for a lower portion of the map (where the high values of these features coincide).

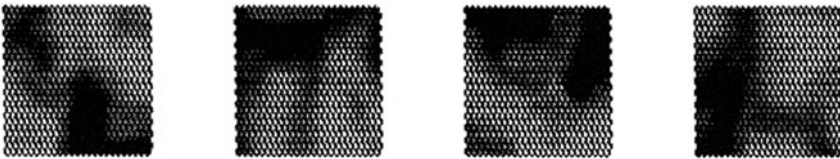


Figure 8. Distribution of features in the SOM (darker regions correspond to higher values of the respective feature of the patterns).

It should be mentioned that for smaller maps, their ability to distribute the classes in disjoint regions was very limited. For instance, in the 5 by 5 map, we were not able to delineate clearly separable regions. Furthermore far more overlap between the classes occurred even for the same neuron. This was a clear indicator of the size of the network not being fully adequate to the size of the data set.

In the sequel, we consider two commonly studied machine learning data sets such as wine and glass data and visualize the performance of the SOMs in these cases.

Wine data This dataset consists of 178 data points (patterns) belonging to 3 classes. We start with a small 5 by 5 SOM which was trained for 2,000 epochs. The patterns

are normalized linearly. The results are shown in Figure 9. By visually inspecting the map, we see clearly identified boundaries that potentially delineate the patterns belonging to different classes. This indeed what has happened here. As the software environment is highly interactive so that we can directly look under the “hood” of the map and inspect the patterns associated with the selected nodes of the map. The homogeneous regions (the nodes with light shading) and their correspondence with the classes are included in Figure 9. Interestingly, the homogeneous regions correspond quite well with the areas of the map of high density of data points (again this effect is visualized in Figure 9(a)).

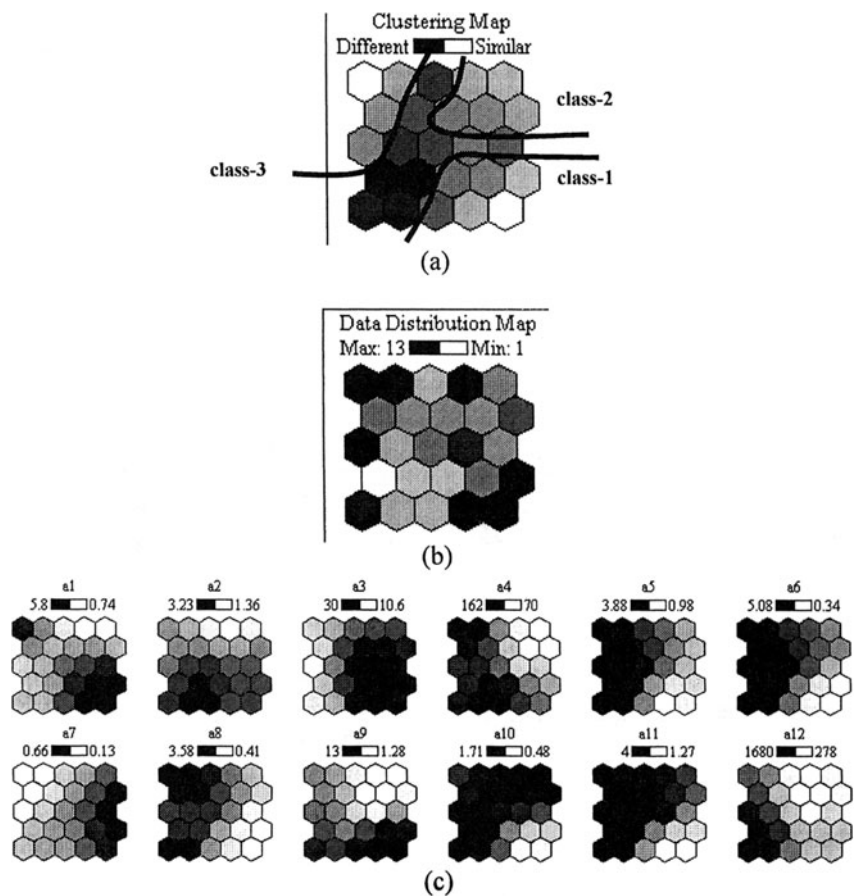


Figure 9. Visualization of data in the SOM with inspection of classes (a) distribution of patterns across the map (b) and individual features distributed across the maps.

This analysis gives us an immediate visual insight into the complexity of the problem treated as a potential classification task. The weight maps reveal dependencies between the features in a graphical form. It can be viewed as a generalization of the standard correlation analysis when we characterize a (linear) dependency between features by a single numeric value (correlation coefficient) while now we are provided by a series of maps one can visually inspect and “correlate”. The details are shown in Figure 9(c). It becomes apparent that some features (shown in the maps denoted by a5, a6, a10, and a11) exhibit the same behavior while others are quite distinct.

Glass data In this experiment, we are concerned with 214 instances of glass belonging to 7 classes. The classification was motivated by criminological investigation. Each pattern is characterized by a number of features dealing with the chemical content of glass (sodium, aluminum, silicon, barium, etc). We start with a 5 by 5 SOM trained for 2,000 learning epochs, Figure 10. Linear normalization is used to preprocess the data.

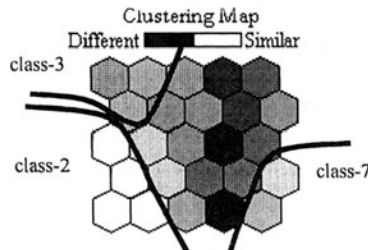


Figure 10. Visualization of glass data (classes) supported by SOM.

Few classes can be delineated, namely 2, 3, and 7. The rest are difficult to distinguish. The increase of the size of the map helps alleviate the problem. With the increase of the size of the map to a 13 by 13 grid, Figure 11, after 7,000 learning epochs we end up with several homogeneous regions identifying most of the classes existing in the problem (the size of the map was made quite large on purpose with an intent to see how far the discrimination between the classes can be realized). The results point out that some classes are easy to discriminate (those are the classes we were able to find in the map) while others such as class require more attention when building their classifiers. Still at this size of the map we were not able to find a clearly distinguished region occupied by class-6. The distribution of the classes in the map reflects the diversity of the patterns belonging to the corresponding class; apparently class-5 is more “compact” than class-1. The mutual distribution of the classes is another interesting indicator as to the relationships between the classes; for instance class-1 and class-2 are neighbors while class-7 is located quite distant from these two. Interestingly, the distribution of patterns across the map is quite uniform,

Figure 11 (b) meaning that all nodes of the map were involved in the organization of the data to a similar extent.

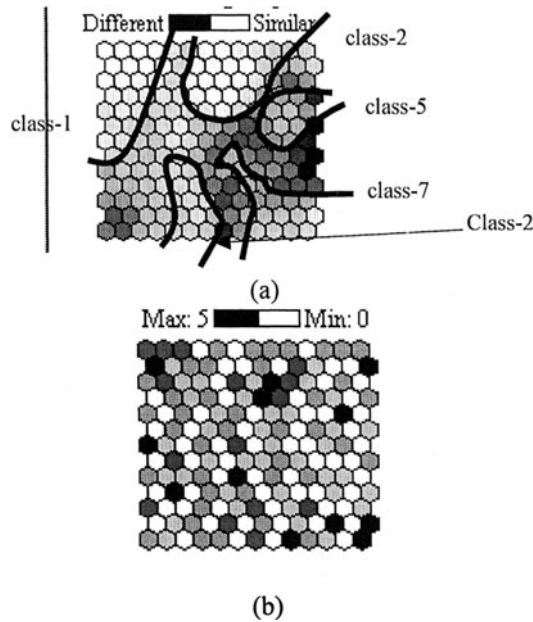


Figure 11. Distribution of classes in the SOM (a) and data density across the map (b).

In both cases, each homogeneous region in the map identified by the designer or data analyst can be directly used towards the construction of information granules. When it comes to intervals, for each variable (feature) we determine the lowest and highest value in the set that form the bounds of the interval. In case of forming information granules in the form of fuzzy sets, we may consider the collection of the algorithms studied in Part III.

15. 5 CASE STUDY: ANALYSIS OF SOFTWARE QUALITY VIA SOFTWARE MEASURES

Software Engineering is concerned with the design of high quality software products and establishing sound development processes. Software quality is difficult to assess. Software measures (Briand et al., 1996; Chidamer and Kemerer, 1994; Li and Henry, 1993; Weyuker, 1988; Zuse, 1985) are attempted to relate to the quality of the software products. In this case study, we discuss how the software data can be analyzed within this context.

Software Measures

Software measures are quantifiable and scalar descriptions of properties of software artifacts. While a comprehensive discussion of the definition and the role of software measures is outside the scope of this study, the readers can refer to the outstanding volume of Fenton and Pfleeger (1997). In what follows, we touch the issue to the extent it help understand the essence of the problem and follow the main flow of data analysis.

Our selection tries to define for this exploratory research a manageable set of measures, yet widely used, and representative of a large number of measurement programs and research work. We focus on Object Oriented metrics given the relevance that Object Oriented techniques have assumed in the last few years. The extension of the approach to other methods does not present any conceptual problem, and is an interesting subsequent step along this line of data analysis.

The Object Oriented measures used in this study come from the collection of software measures proposed by Chidamber and Kemerer (1994): Coupling Between Objects, CBO – the number of distinct classes associated to the target class, Depth of Inheritance Tree, DIT – the length of the longest path to a superclass; Lack of Cohesion between Methods, LCOM – the difference between empty and non-empty intersections between sets of attributes that are accessed by different methods; Number of Children, NoC; Response For a Class, RFC – the number of different methods that can be executed in response to the call of a method of a class, WMC, Weighted Method per Class – the weighted number of methods in the class, in our case we follow the “usual” approach and we assign unitary weight to the classes (Li and Henry, 1993).

We are concerned with 5 C++ and 5 Java projects composed of a significant number of classes described in terms of the already discussed software measures. The projects have been collected from public domain systems available through the WWW. The use of C++ and Java projects in the same study is also helpful in comparing procedural and object-oriented software development environments.

Visualizing Relationships Between Software Measures with SOMs

In this experiment, we confine ourselves to the maps formed as grids of 10 by 10 neurons. The number of learning epochs is equal to 2,000. The learning rate α is set up originally as equal to 1 and then it is reduced exponentially to 0.01 at the end of the learning process. The initial neighborhood radius starts from 3 and it becomes reduced linearly to 1. The frequency sensitive coefficient ϵ is equal to 0.01. Furthermore the software measures are linearly normalized.

Once the SOM has been learned, then the map can be used as a basic vehicle to visualize data. We can envision two main ways of data representation and a visualization of the underlying topology of the data set, namely

1. Visualization of the relationships between the classes on the map
2. Visualization of the relationships between the measures in the map

Visualization of the relationships between the classes on the map. This is a common way in which SOMs are used in many areas of data analysis. Each data point is a vector of the software measures used in the experiment. This vector is presented to the already optimized SOM and the winning node is determined. Then the data index is assigned on the corresponding entry of the map where the winning node has been located. This process is repeated for all data. What we end up with, is a collection of data indexes distributed across the map. They clearly illustrate what the groups of data are, how they are distributed, and how large or compact they are. Moreover, we visualize how the clusters are located in terms of their vicinity, which may give us an impression as to their potential interaction. The use of the SOM in this capacity is visualized in Figures 12 and 13 for selected JAVA and C++ projects.

For instance, it is obvious, refer to Figure 12, that software classes 468 and 643 (both in right-bottom corner of the map) are very *similar* whereas the two other classes, 468 and 354, (right-bottom vs. top-left corner) *differ* quite substantially. This type of visualization helps us to form clusters of classes that are similar (in the sense of their software measures' manifestation).

The collection of clusters can be an interactive process. One can easily delineate larger clusters (the process of grouping them may be user-interactive) and analyze their properties. The cluster encircled in the map and denoted by "A" consists of classes that have keyword `error` in their description. These classes have the same value of DIT equal to 1. Essentially, those are classes dealing with error handling. The other cluster, denoted by "B", involves classes that are inherited from handler; all of them have the same values of the software measures, that is DIT = 2, NOC = 0, and CBO = 2.

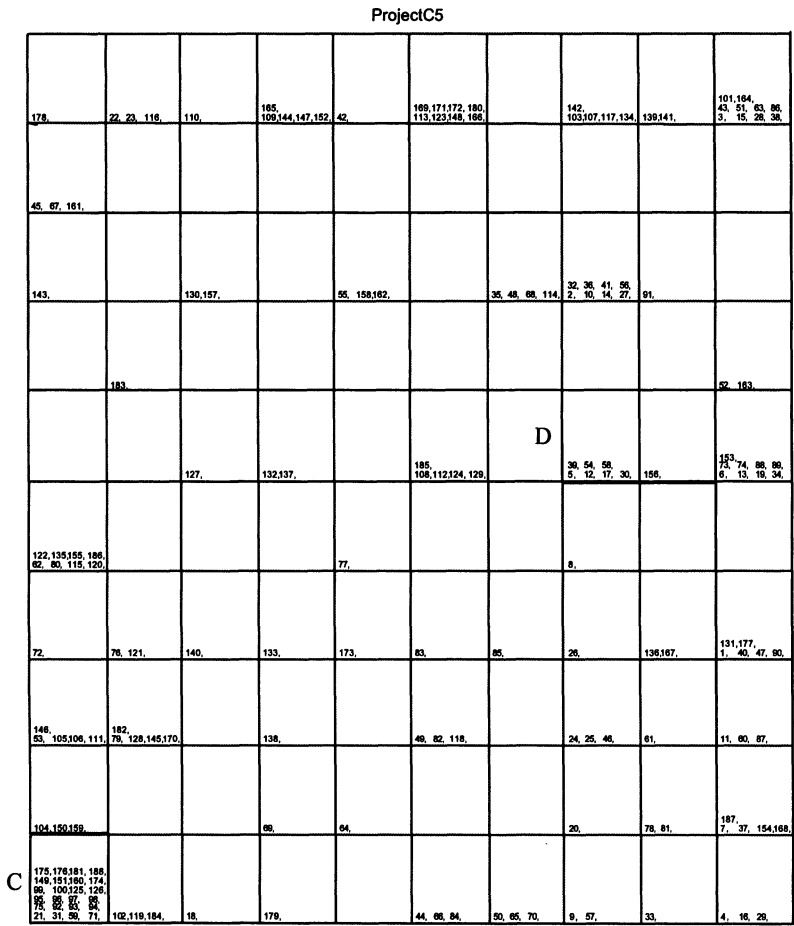


Figure 13. A two-dimensional SOM visualization of C++ software project ProjectC5 (ProjectC5 is a free, multiple platform C++ graphical user interface framework designed to help develop C++ GUI applications. It has 184 classes).

The same type of analysis can be applied to the C++ software project in Figure 13. Cluster "C" consists of classes coming from a corporate include directory. They have the mean value of LOC=2.5 and all other metrics are zero indicating that these

are all very small components. The other cluster, denoted by "D", involves classes dealing with windows and control layout. Typically, these classes have keyword `CanvasPlane` in their description. All of them have the same values of software measures $DIT = 3$, $NOC = 0$.

There are several possible ways of using the clustering information gathered from SOMs.

Clusters can be indexed by the "typical" values of the measures. There is a limited, but still significant, body of knowledge of relations between internal software measures and external property of the resulting product. Knowing that classes of a given group – for instance for project J4 the local includes, end up in a specific cluster where the measures have predefined values, supports understanding the expected behavior of the portion of the product built by such classes. In this way clustering information can be used prior to the launching of the project to predict the overall development effort of the system.

The overall distribution of clusters can be used to catch some of the affinities between software projects, and then use such affinity to predict future behavior based on experience.

In most of the cases the relationships between measures and external product attributes is unknown. When the information about such external product attributes is known, we can use SOM to find new relations, especially those non-linear relations that are hard to find using the standard statistical tools.

We have carried out the training of SOMs for three distance functions. It has been found that the results are very similar as to the mutual location of the software measures in the maps (note, however, that the maps are not identical as the regions may be usually rotated however the relationships between them remain practically the same).

15. 6 CASE STUDY: A GRANULAR ANALYSIS OF ECG DATA

The complex problem of computerized diagnostic classification of the electrocardiogram (ECG) signal is considered as a real world example. It uses a CORDA database, developed by J. Willems at the Medical Informatics Department of the University of Leuven (Willems et al., 1987) consists of 3253 12 lead ECGs (2140 men and 1113 women with a mean age of 49 ± 12 years). There were 12 standard leads (that is I, II, III, AVR, AVL, AVF, V1, V2, V3, V4, V5, V6). It consists only of single-disease cases with normal QRS duration and no conduction abnormality. Seven diagnostic classes have been considered: normal (N), left ventricular hypertrophy (LVH), right ventricular hypertrophy (RVH), biventricular

hypertrophy (BVH), inferior (IMI) anterior (AMI) and combined (MIX) myocardial infarction.

From the original ECG signal (12 standard leads acquired at 500 Hz for a period of about 10 seconds) a set of 540 (45 for each lead) primary measurements were computed with a computerized system, obtaining a first consistent data reduction. A second data reduction, according to a clinical selection and a statistical selection, has been performed obtaining a set of 39 ECG features. They include amplitudes and duration of the QRS and T waves, QRS and T axes, ST-segment elevation or depression, and the area under QRS and T waves.

The same dataset has been used to establish the performance of statistical classification models (Willems et al., 1987) and to validate the performance of different architectures of neural networks (Bortolan and Willems, 1994; Silipo et al., 1999).

One can envision a certain hierarchy of the classes of the signals that could be helpful in understanding the results of self-organization. The diagnostic class of biventricular hypertrophy (BVH) then includes both LVH and RVH, and consequently the three classes BVH, LVH and RVH are not completely independent. This means that a classification of LVH + RVH is equivalent to BVH, and that a BVH patient classified as only LVH or RVH represents a partial discrepancy. Analogous considerations are valid for the diagnostic class of combined myocardial infarction MIX with respect with AMI and IMI.

The size of the map was experimented with. Finally, the size of 25 by 25, Figure 14 comes as a reasonable choice considering the size of the data set as well as the interpretation results one can derive (it is worth noting that this description of data is an interactive process so the user has control over the granularity of the descriptors visible through the map). The data were normalized with the use of a logistic transformation.

There are of immediate and important observations one can make on the basis of a visual inspection of the self-organizing map (especially the region map and the maps of the individual features). We may quantify the groups in a more quantitative manner as summarized in Table 1. These groups of data are described in terms of class homogeneity, total size, and fuzzy sets – information granules capturing the data beneath the selected portion of the map.

Several interesting observations can be drawn

- The homogeneous regions in the SOM, their size and location vis-à-vis other regions help identify relationships between the classes of ECG signals.

In particular, a region (cluster) capturing normal signals (region C in Figure 15) is quite compact and shows a high level of homogeneity. The region

denoted by A (that involves AMI) is quite extended and is quite distant from other regions. Similarly, region B (that captures a mixture of IMI and MIX) is apart from the other regions and occupies an entire region on the upper right corner of the map. A very different behavior can be observed for the three other regions, that is E, H, and F. These are close neighbors and all of them capture two classes RVH and BVH but in a different mix. When moving along the map and starting from the first one (E), there is an evident mix of BVH and RVH. In the sequel, the next group (H) is dominated by RVH while the group identified as F has a similar dominance by RVH with some BVH.

- The identification of the groups in the map can be viewed as a descriptive data analysis with an ultimate goal to capture the essence of the data. In this case we are interested in building concise and homogeneous descriptors of the ECG classes. The map tells us what is most likely as to the occurrence of “plain” or mixed classes of patterns. Obviously, it is easy to describe (and discriminate) between the class of normal signals (N) and others while discriminating between class IMI and MIX (as shown in region B) will be a difficult task (no matter what classifiers we are interested in). It is easy to discriminate between class RVH when dealing with region H however doing the same for the region E (where there is an evident mix of RVH and BVH) will be a significant classification challenge.

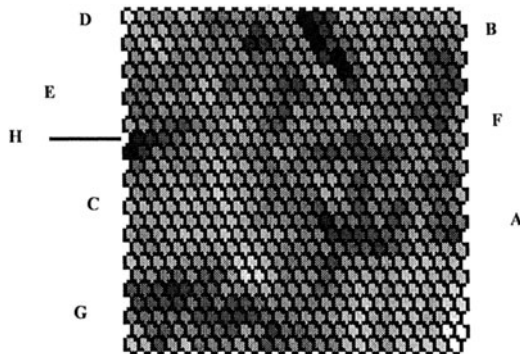


Figure 14. The self-organizing map (size of 25 by 25) and several clusters identified for further analysis.

The series of maps for each feature (feature maps) as shown in Figure 16 is important source of information that helps us visualize relationships between the features of the signal. A quick visual inspection helps us notice that some of them are highly related (the corresponding maps are very similar). For instance,

- the parameters A21 (Q amplitude in V3) and A22 (Q duration in V3) as well as partially A19 (Q duration in V1) exhibit similar behavior, showing a region with high values in the upper right corner, with a correspondence (in agreement) with the classification of region B.
- some qualitative similarities can be seen considering the parameters A27(ST elevation in V6), A28 (ST slope in V6), and A10 (ST elevation in II).
- a qualitative similarity is shown by A14 (area under T wave in lead AVR) and A13 (ST elevation at 80 ms after J point of lead AVR)

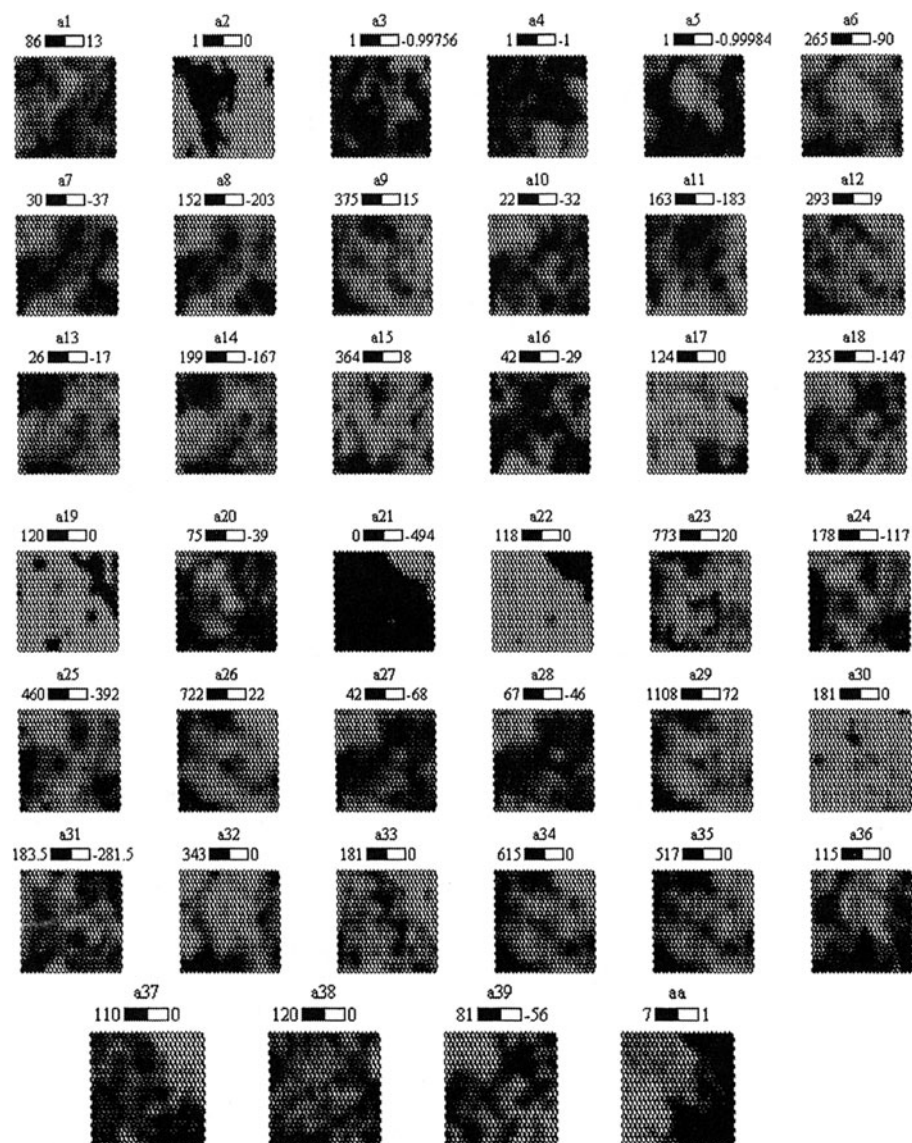


Figure 15. Feature maps of the ECG patterns; the features are denoted as a1, a2, ..., a39. The brightness scale shows the values of the features as they are distributed across the map.

Figure 16 displays a density map illustrating how the ECG patterns populate the SOM. In general, the data become distributed across the map quite uniformly with an exception of few entries. Nevertheless the differences are not very substantial. The density map states that there are no any problems with the learning as there were no particularly “hyperactive” neurons during the learning process.

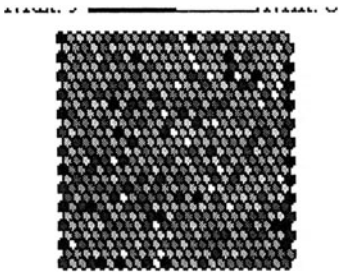


Figure 16. Density map associated with the SOM – the brighter the color the higher the number of the data points allocated to the corresponding neuron. The range of these numbers is from 9 (the darkest location of the map) to 0 (the brightest entries of the map).

Region of the map	Number of patterns	Homogeneity (number of patterns across classes)	Description of the region
A	248		Occupied by class “AMI” with some patterns from class “MIX”
B	93		Class “IMI” and “MIX “ are represented in almost equal mixture
C	119		Clas “N” dominates this cluster

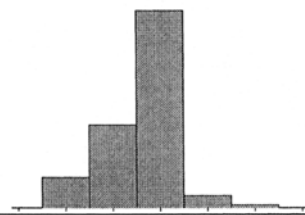
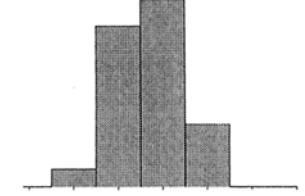
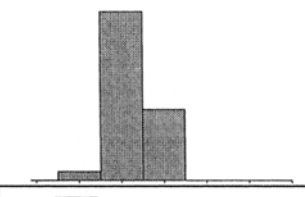
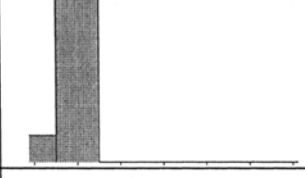
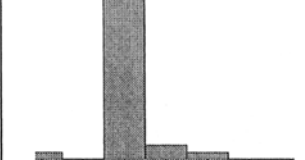
D	106		Class "BVH" with some data coming from class "RVH"
E	48		Class "BVH" and "RVH" in an almost equal mix with a few patterns in class "LVH" and "AMI"
F	28		Class "RVH" dominates here with some patterns coming from class "BVH"
G	23		Class "LVH" with a few patterns in class "N"
H	28		Class "RVH" with few patterns belonging to class "BVH"

Table 1. Characteristics of the granular descriptors of the ECG classes.

15. 7 CONCLUSIONS

Data analysis cast in the setting of information granules arises as an interesting and attractive option of data exploration. This chapter has concentrated on the use of self-organizing maps as a vehicle of user-centric interactive data analysis. It is worth

underlining that while the data analyst identifies homogeneous regions of the map that are afterwards converted into information granules, this needs to be treated as a preliminary phase of any model design. We can view these granules as a blueprint (conceptual skeleton) of any model. The size of the granules over which it is formed implies its specificity/generality and a level of necessary details one intends to capture within the model. In the sequel, the model is subject to further parametric optimization.

REFERENCES

- Bargiela, A., Pedrycz, W. (2001), Classification and clustering of granular data using SOM, *IFSA-NAFIPS 2001*, Vancouver (BC), July 2001, 1696-1701.
- Bortolan, G., Willems, J.L. (1994), Diagnostic ECG classification based on neural networks, *Journal of Electrocardiology*, **26**, 75-79.
- Briand, L.C., S. Morasca, V.R. Basili (1996), Property-based software engineering measurements, *IEEE Trans. on Software Engineering*, **22**, 68-86.
- Chidamber, S.R., C.F. Kemerer (1994) A Metrics suite for object-oriented design, *IEEE Transactions on Software Engineering*, **20**(6).
- Fenton, N.E., S.L. Pfleeger (1997), *Software Metrics: A Rigorous and Practical Approach*, PWS, London.
- Kohonen, T.(1982), Self-organized formation of topologically correct feature maps, *Biological Cybernetics*, **43**.
- Kohonen, T. (1995), *Self-organizing Maps*, Springer Verlag, Berlin.
- Kohonen, T., S. Kaski, K. Lagus, T. Honkela (1996), Very large two-level SOM for the browsing of newsgroups, In: *Proc of the Int Conf. on Artificial Neural Networks*, Bochum, Germany.
- Li, W., S. Henry (1993) Object oriented metrics that predict maintainability, *Journal of Systems and Software*, **23**(2)
- Oja, E., Kaski S. (eds) (1999), *Kohonen Maps*, Elsevier, Amsterdam.
- Silipo, R. Bortolan, G., Marchesi, C. (1999), Design of hybrid architectures based on neural classifier and RBF pre-processing for ECG analysis, *Int. J. of Approximate Reasoning* **21**, 177-196.
- Willems, J.L., Lesaffre, E., Pardaens, J. (1987), Comparison of the classification ability of the electrocardiogram and vectorcardiogram, *American J. Cardiology*, **59**, 119-124.
- Weyuker, E.J. (1988) Evaluating software complexity measures, *IEEE Transactions on Software Engineering*, **14**(9).
- Zuse, H. (1985) *A Framework of Software Measurement*, de Gruyter, Berlin.

TEMPORAL GRANULATION AND SIGNAL ANALYSIS

16. 1 INTRODUCTORY NOTES

In this chapter, we discuss an application of information granules in the description of time series and signal analysis, in general. Signal analysis is predominantly numeric (as sources of data related with physical phenomena). Sensors generate millions of readings. With the growing interest in carrying out high-level conceptual description of signals and/or develop efficient algorithms of signal processing, it becomes evident that we need to explore information granulation more vigorously and extensively. As a matter of fact, digital signal processing is omnipresent today. A number of pursuits hinge upon information granulation and resulting information granules. Syntactic pattern recognition of signals (Horowitz, 1975; Kundu et al., 2000; Papakonstantinou et al., 1986; Papakonstantinou, 1981; Skordolakis, 1986; Trahanias and Skordolakis, 1990; Udupa and Murphy, 1980) operates on information granules – a collection of segments of signals that build an alphabet of symbols on which the processing (grammar inferences) are accomplished. Similarly, data mining of time series (Cios et al., 1998; Bargiela, Pedrycz, 2002) operates on segments of signals and attempts to make sense of their sequences; in this sense the pursuits in this realm heavily depend on information granules of some kind.

In general, signals are granulated in two dimensions: time and amplitude (space). The granulation of time and space can be realized in many different ways, see Figure 1. The one realized in time is referred to as sampling. The samples of the signal can be collected together through some temporal windows (Hamming, exponential, uniform, etc.). The commonly used spatial granulation took advantage of interval granulation where such granulation is realized in the realm of analog-to-digital (A/D) conversion.

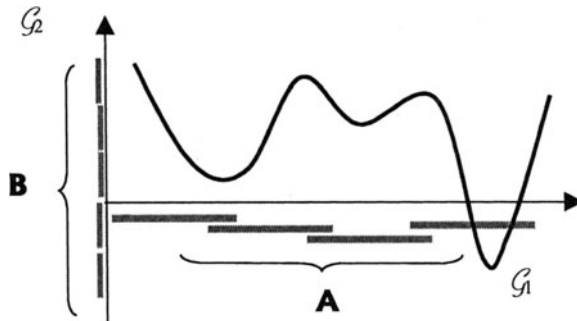


Figure 1. Signal granulation in time and space (amplitude) with a variety of granulation formalisms (G, I, K) and families of generic information granules (**A**, **B**, **C**).

We concentrate on the granulation using fuzzy sets, show pertinent algorithms and then discuss models of signal processing. The experimental part is devoted to two selected applications of granular data to signal processing such as granular predictive models and an idea of condensation of numeric signals and their graph representation. We consider synthetic as well as real-world data sets. The first ones help illustrate the underlying idea. The second group of data sets comes the MIT-BIH database of ECG signals.

16. 2 GRANULATION OF SIGNALS IN SPATIAL DOMAIN

Here we discuss the process of granulation of signals in the setting of fuzzy sets. The construction of meaningful fuzzy sets is governed by the criterion of experimental justification of information granules and their maximal specificity (highest granularity).

The Development of Data-Justifiable Information Granules: A Formulation

Our key objective is to construct fuzzy sets that are legitimized by data. The problem is posed in the following manner

-given is a collection of numeric one-dimensional data $\mathbf{X} = \{x_1, x_2, \dots, x_N\}$ where x_k is a real number. Construct a fuzzy set **A** belonging to a certain family of fuzzy sets **A** (say, triangular, parabolic, etc.) so that it "legitimized" (experimentally justified) in the sense of its experimental evidence while being *specific* enough so that its support is kept small.

The above formulation of the problem has a strong intuitive underpinning. On one hand, we want the fuzzy set to embrace enough experimental evidence. On the other

hand, the information granule should become specific enough. These two requirements, that are conflicting to some degree, can be articulated in the following manner

- maximize the sum of membership values
-

$$\sum_{k=1}^N A(x_k)$$

(note that the above is just a probability of the fuzzy event A manifested through the discrete data set X , see Chapter 3. The maximization of this sum implies that we make the information granule highly justifiable from the experimental standpoint)

- minimize the support of the fuzzy set that leads to higher specificity

$$\text{measure}(\text{supp}(A)) = (b-a)$$

where "a" and "b" are the bounds of the support of A . Refer to Figure 2 that illustrating the character of these two requirements along with their conflicting nature: the fuzzy set in Fig 2(a) is very "specific" yet it does carry a very limited experimental evidence (note a limited number of data "embraced" by the fuzzy set). In contrast, Figure 2(b) reveals an opposite situation: we have an information granule of a large size (not being specific) but supported by a significant number of data points.

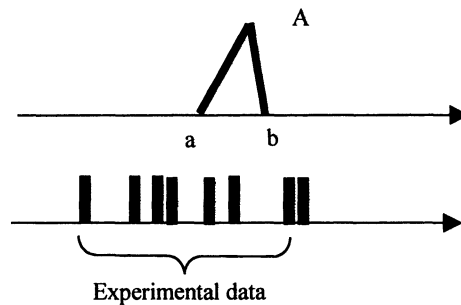


Figure 2(a). Information granules-fuzzy sets satisfying one of the optimization criteria; see a detailed description in the text.

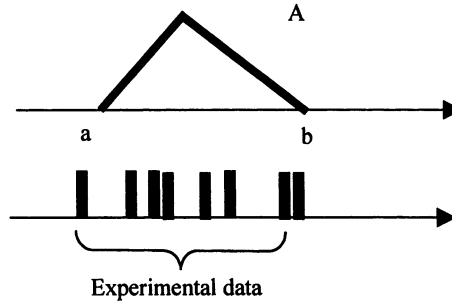


Figure 2(b). Information granules-fuzzy sets satisfying one of the optimization criteria; see a detailed description in the text.

We can combine these two in a form of a single index Q being a ratio of these two

$$Q = \frac{\sum_{k=1}^N A(x_k)}{\text{measure}(\text{supp}(A))} = \frac{\sum_{k=1}^N A(x_k)}{(b-a)} \quad (1)$$

Apparently, in light of the above requirements, Q has to be maximized

$$\max_p Q \quad (2)$$

with p denoting a collection of the parameters of the membership function to be optimized.

16. 3 The detailed granulation algorithm

In what follows, we confine ourselves to modal fuzzy sets. This helps us design a general algorithm while not losing generality of the resulting construct. We can envision that the property of modality is a highly desired property retaining the semantics of the information granule. Moreover the modal nature of the membership function helps us handle the development of the fuzzy set by looking into its decreasing portion and increasing portion separately.

The process of the formation of the information granules can be split into two phases

- determining the numeric representative of the data set
- building the detailed membership function (here we assume that its form is given a priori)

In the first phase, we may consider the numeric representative of the data to be the first, quite rough descriptor of X . This phase does not involve granular entities at all

but alludes to a numeric "compression" of the data set. In this case numerous, well-known methods exist: a mean value, median, etc. Anticipating the second, more refined phase, our choice is to proceed with the median. Median is a robust estimator so its value does not depend on any outliers (the property that does not hold for the mean value). The calculations of the median are also straightforward. If \mathbf{X} is ordered, the median splits the data set in halves. If \mathbf{X} is unordered, the median (med) is a solution to the following L_1 -optimization problem

$$\min_m \sum_{k=1}^N |x_k - m| = \sum_{k=1}^N |x_k - \text{med}| \quad (3)$$

The median is taken as the modal value of the fuzzy set. This splits the data into two subsets that are processed separately leading to the computations of the left-hand and right-hand portion of the membership function of A . The parameters of A are then determined separately. In some sense, we can view the second phase of the formation of the information granules as a refinement of the "compression" scheme already initiated by the numeric representative of \mathbf{X} (that is its median). Here, two main approaches can be exercised, Figure 3:

- data-driven: we select the values of the parameters of the membership function based on the finite number of data meaning that they assume some discrete values implied by the original data
- optimization approach

The first method is straightforward and does not require any excessive optimization effort. We sweep through all data points considering each of them to be a potential value of the parameter of the membership function (cutoff point, that is denoted here by "a"). The one that maximizes the performance index

$$Q(a) = \sum_{\substack{k=1 \\ x_k < \text{med}}}^N A(x_k) / (\text{med} - a)$$

forms the solution to the problem $Q(a_{\text{opt}}) = \max Q(a)$. Note that in the above formulation we were dealing with the increasing portion of the membership function. Evidently, the same process is carried out for the decreasing portion of the fuzzy set.

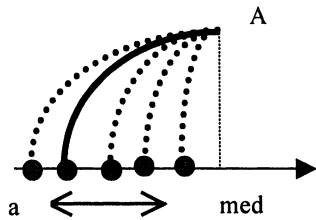


Figure 3. Detailed computing of the parameter (cutoff point a) of the fuzzy set.

The second approach, that is a fully-fledged optimization method, maximizes Q over all possible values of " a ". This process could lead to the higher values of the performance index yet it comes with more profound computational overhead.

As a numeric illustration, we discuss a synthetic data set is shown in Figure 4 which represents a discrete time series.

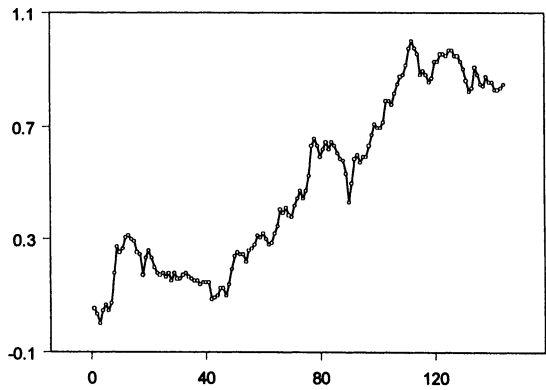


Figure 4. A synthetic time series.

To illustrate the underlying optimization process, let us consider a section of 20 successive samples (granulation window) of the synthetic time series, Figure 5. The granulation is realized with the use of the parabolic fuzzy set (parabolic membership function).

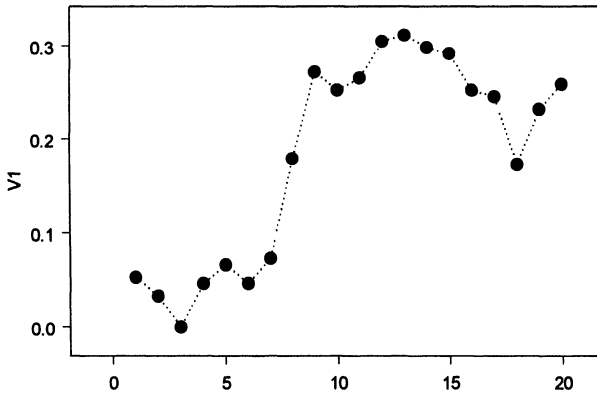


Figure 5. Data to be granulated - a segment (granulation window) of the entire data set.

The computed median of this granulation window is equal to 0.245033.

Determining the bounds of the parabolic membership function, we get the following values of the cutoff points

- using the direct method these are equal to 0.00 and 0.245, respectively. Note that the range of the amplitude of this segment of the time series is equal to [0.000000, 0.311258]
- optimizing the cutoff points, we obtain the values equal to 0.00 and 0.257. These are different and produce slightly higher values of the performance index Q.

The plots of the performance index obtained for the two methods are illustrated in Figures 6 and 7.

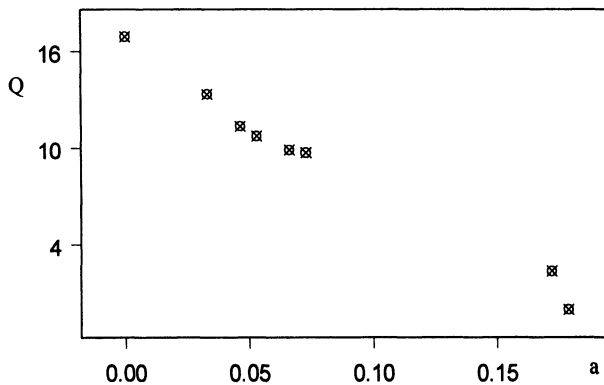


Figure 6(a). Performance index Q versus the lower bound of the fuzzy set: direct enumeration.

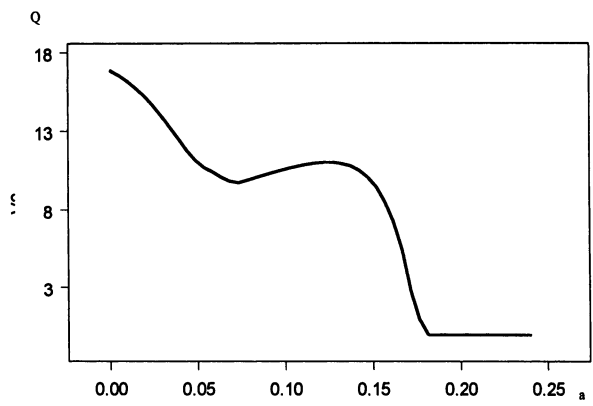


Figure 6(b). Performance index Q versus the lower bound of the fuzzy set: optimization.

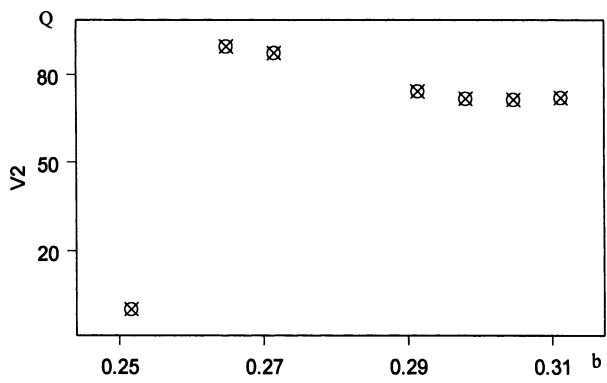


Figure 7(a). Performance index Q versus the upper bound of the fuzzy set: direct enumeration.

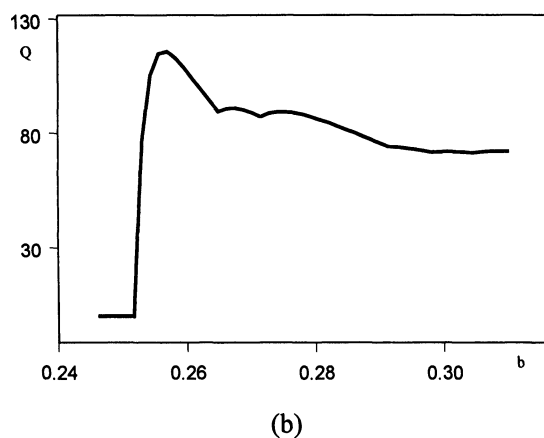


Figure 7(b). Performance index Q versus the upper bound of the fuzzy set: optimization.

In the sequel, we consider another granulation window coming from the entire data set and illustrated in Figure 8.

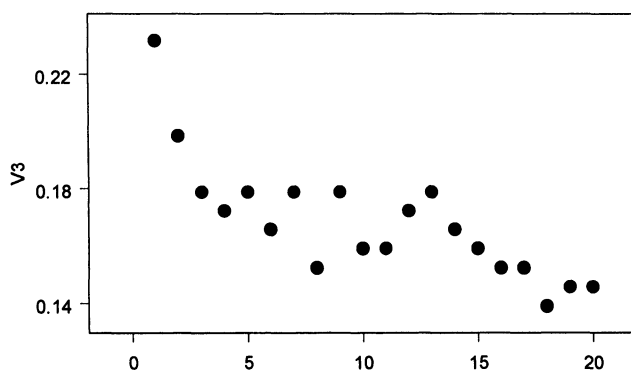


Figure 8. Data to be granulated.

This segment of data is spread between 0.139 and 0.232 with the median equal to 0.1655. The results of computing the bounds of the parabolic fuzzy set are contained in Figure 9 and 10.

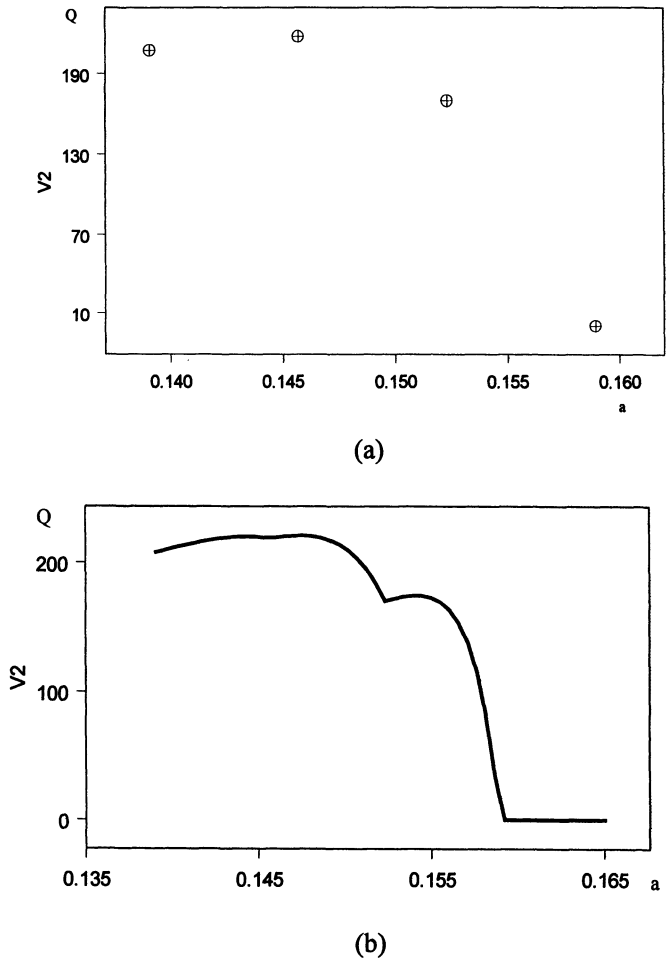


Figure 9. Performance index Q versus the lower bound of the fuzzy set: direct enumeration (a) and optimization (b).

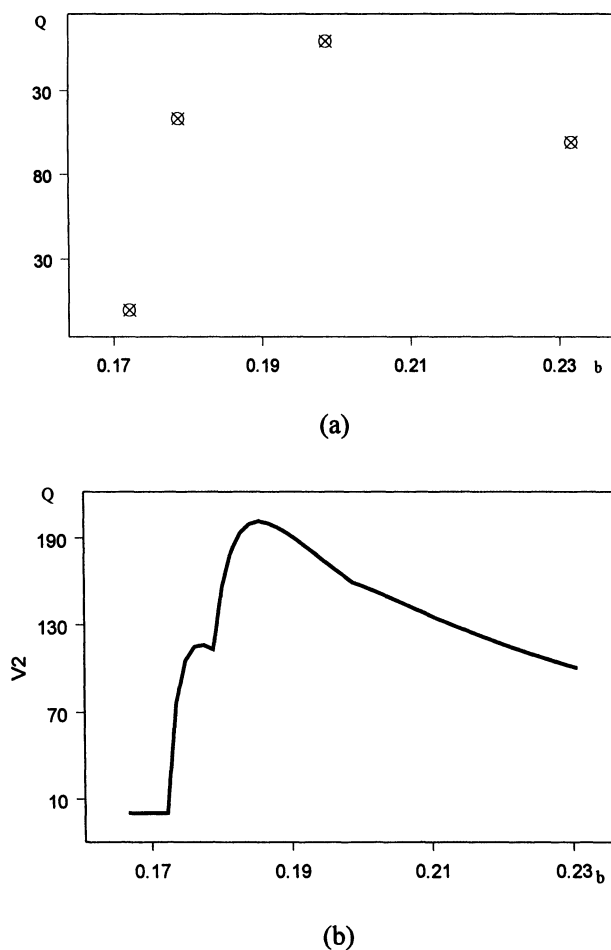


Figure 10. Performance index Q versus the upper bound of the fuzzy set: direct enumeration (a) and optimization (b).

16. 4 GRANULAR MODELS OF SIGNALS

Signal processing is predominantly numeric. Numeric data are processed in a linear or nonlinear fashion. Once we get into information granules, they give rise to a new dimension of signal analysis and signal processing. Interestingly, some of already existing pursuits of signal analysis and classification such as syntactic pattern recognition dwell on symbolic elements. In this section, we elaborate on two interesting ideas that directly originate from the framework of granular computing.

Predictive Description of Granular Models

The first-order linear systems are in common usage. We expand on it by proposing a first-order granular dynamic model linking the actual information granule with the predicted one. The underlying formula reads as follows

$$B = A \oplus \partial A T$$

Where A and ∂A are the information granules of the signal describing its current status (namely, an amplitude A and the trend of the granule described by the first-order derivative, ∂A). Both A and ∂A are determined as fuzzy sets (triangular, parabolic, etc.) based on the numeric data contained in the granulation windows. More specifically, A is obtained through the direct enumeration of the data $\{x_1, x_2, \dots, x_N\}$ whereas ∂A originates from the same construction applied to $\{x_2-x_1, x_3-x_2, \dots, x_N-x_{N-1}\}$. The size of the temporal granule is denoted by T . The above expression is a shorthand symbolic notation and requires some clarification: First, the operation of addition need to be treated in a sense of addition of two fuzzy numbers (with given membership functions). Second, the multiplication occurring above is completed for a single numeric value (T), the result is easy to compute and this multiplication does not affect the form (class) of the membership function. As a matter of fact, it realizes a simple scaling process. The size of the temporal granule (T) modulates a level of impact of the changes (∂A) on the predicted information granule and is subject to some optimization procedure. In other words, we look for an optimal value of T , T_{opt} , so that the predicted information granule B matches the information granule B' manifesting in the time series.

Condensation of Numeric Signals

Naturally, information granules help "condense" the signal and represent it in the form of the sequence of information granules - fuzzy numbers. In a nutshell, this type of condensation moves us up from the numeric level up to the symbolic processing layer. The size of the granules (granulation windows and subsequently fuzzy sets) implies a level of abstraction that is achieved. The level of abstraction is essential in many possible ways. First, we can develop models that capture and articulate relationships at the higher level of abstraction. This leads to models that are easier to understand and which give a better insight into the nature of the phenomenon. The information granules serve as basic building blocks used afterwards in a variety of models. For instance, syntactic pattern recognition dwells on a family of structural elements (Horowitz, 1975). In this case these structural elements are just fuzzy numbers. For each granulation window, we define a fuzzy set capturing the amplitude of the signal and another fuzzy set describing its changes. This leads to the pair $(A(K), \partial A(K))$ (in contrast to the previous notation

in the original space, we use a capital letter to denote that this concerns a different time scale). More descriptively, the granulation process and the ensuing representation can be described as follows

$$\{x(1), x(2), \dots, x(k), \dots\} \Rightarrow \{(A(1), \partial A(1)), \dots (A(K), \partial A(k)), \dots\}$$

While the above representation gives us a certain insight into the sequence of the information granules, they can be connected together in the form of a web of generic entities. More specifically, as the number of the information granules (or their combinations) $A(K)$, $\partial A(K)$ could be high, they are clustered (grouped) and then used as the components of the web. Denote the prototypes (representatives) of the clusters by $(A(1), \partial A(1)), \dots, (A(c), \partial A(c))$ where "c" stands for the number of the clusters. The clustering method is secondary to this problem; a FCM method or its relative could be a plausible choice [1]. The collection of the information granules is then mapped onto the structure of the prototypes. This mapping is realized by determining the connections between the nodes (prototypes), Figure 11.

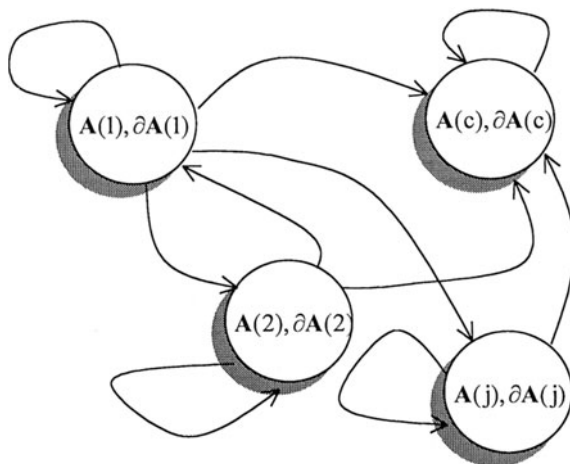


Figure 11. The web of prototypical information granules.

16. 5 EXPERIMENTAL STUDIES

In this section, we proceed the numerical analysis we have already started with. Proceeding with the signal in Figure 4, we get its description summarized in Table 1. The table includes information granules describing the amplitude of the signal as well as its changes, ∂A .

$A(K)$			$\partial A(K)$		
0.000000	0.245033	0.264901	-0.039735	0.006623	0.039735
0.145695	0.165563	0.198675	-0.026490	-0.006623	0.013245

0.145695	0.165563	0.198675	-0.026490	-0.006623	0.013245
0.086093	0.218543	0.278146	-0.006623	0.006623	0.013245
0.384106	0.417219	0.523179	-0.026490	0.026490	0.059603
0.569536	0.602649	0.668874	-0.046358	0.000000	0.066225
0.847682	0.880795	1.000000	-0.026490	0.013245	0.039735
0.821192	0.900662	0.966887	-0.039735	-0.006622	0.013245

Table 1. Information granules (the size of the segments - granulation windows is equal to 20 elements).

The series of Figures, Figure 12 to 16 illustrates the plots of information granules ($A(K)$ and $\bar{A}(K)$) for several selected values of K . The bounds of the parabolic fuzzy sets are marked using a dotted line. An observation is in place: the larger the granulation window, the more synthetic and concise the description becomes while the granules themselves get broader.

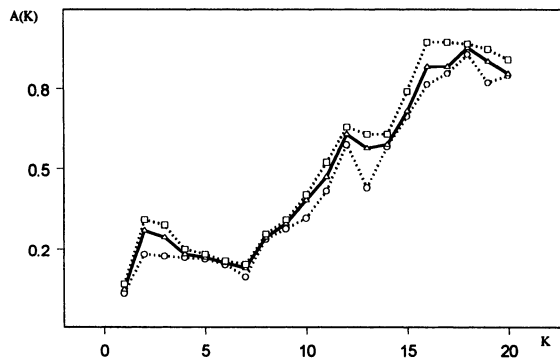


Figure 12. Plots of $A(K)$ (granulation window equal to 7).

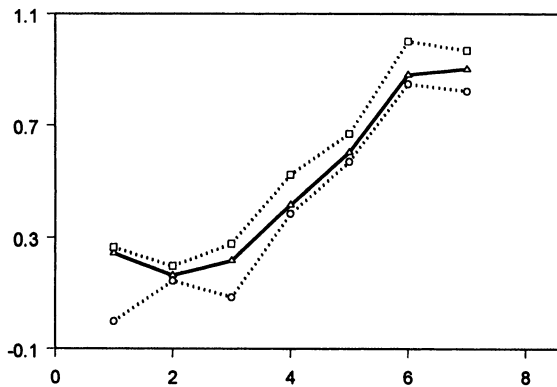
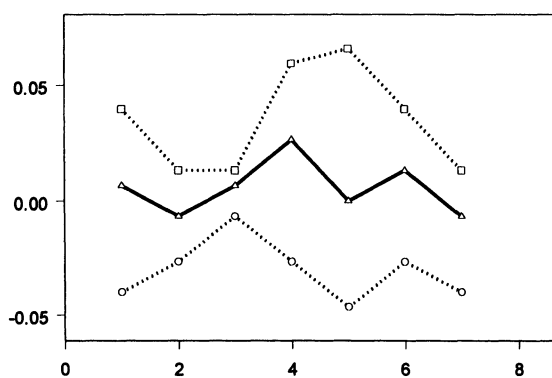
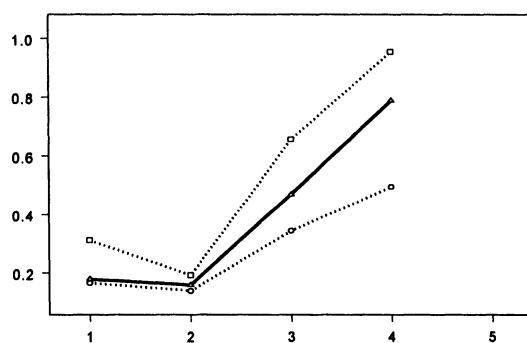
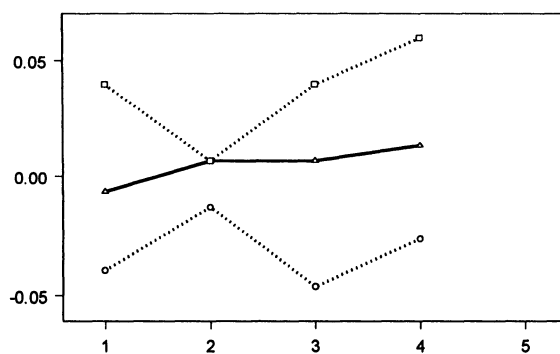


Figure 13. Plots of $A(K)$ (granulation window equal to 20).

Figure 14. Plots of $\partial A(K)$ (granulation window equal to 20).Figure 15. Plots of $A(K)$ (granulation window equal to 30).Figure 16. Plots of $\partial A(K)$ (granulation window equal to 30).

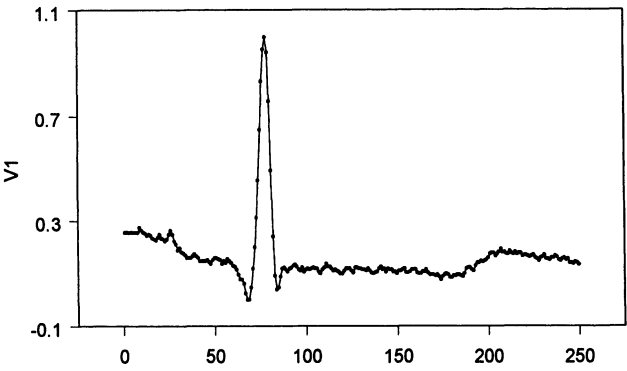


Figure 17. An original ECG signal (a single QRS complex).

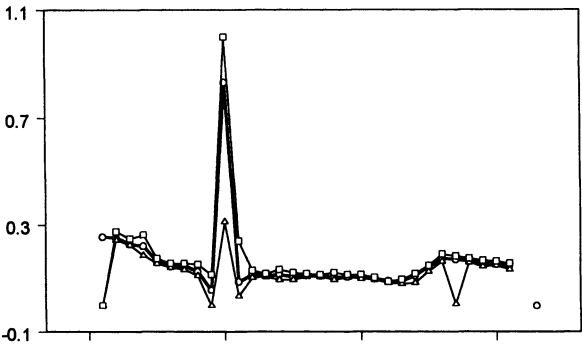


Figure 18. Condensed ECG signal (the size of the granulation window equal to 8).

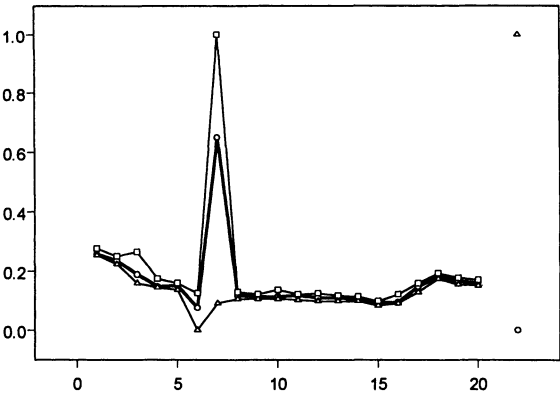


Figure 19. Condensed ECG signal (the size of the granulation window equal to 12).

In the current example, we consider an ECG signal, Figure 17 and proceed with its granulation. By varying the size of the granulation window, various granular representations of the same initial signal are obtained, Figure 18.

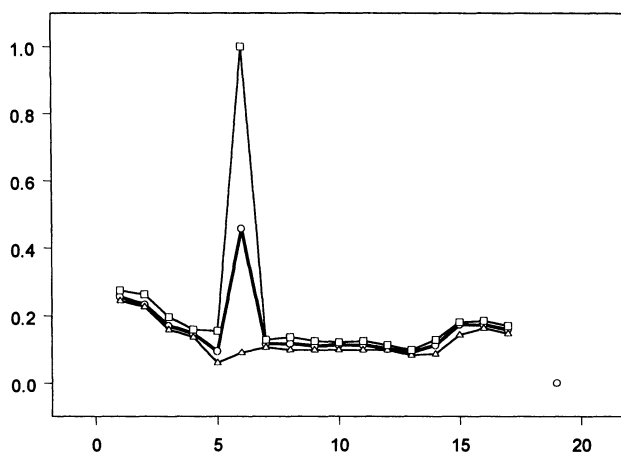


Figure 20. Condensed ECG signal (the size of the granulation window is equal to 14).

Figure 21 illustrates another ECG signal while Figure 22 shows the results of its temporal granulation.

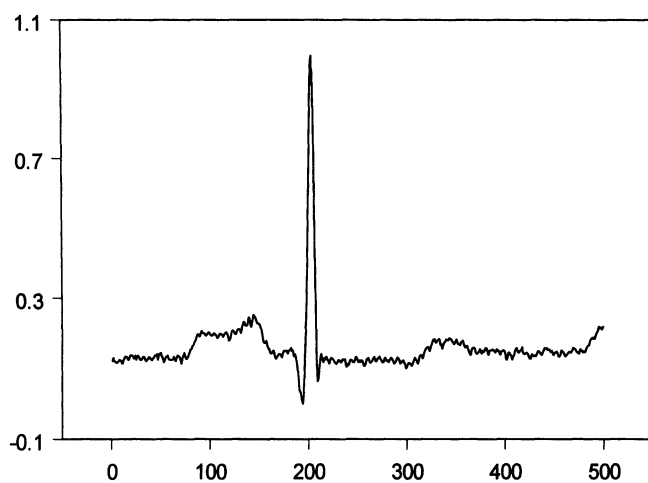


Figure 21. An example ECG signal.

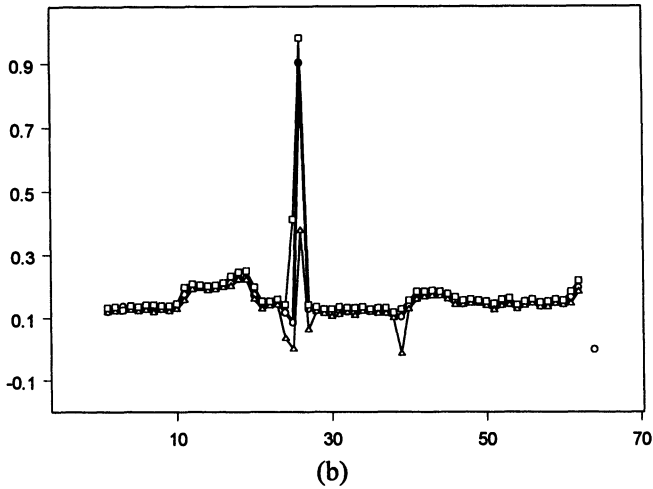


Figure 22. Granulation of the ECG signal; segmentation window is equal to 8.

In the sequel, we exploit the predictive capabilities of the model. We consider two temporal windows of 8 and 9 successive time samples. The performance index (V) expressed as a sum of absolute distances between the parameters of the predicted information granule B and the one (B') resulting from the granulation of the experimental data,

$$V = \sum_K \{ |a(K) - a'(K)| + |m(K) - m'(K)| + |b(K) - b'(K)| \}$$

In the above performance index, $a(K)$, $m(K)$, $b(K)$ and $a'(K)$, $m'(K)$, $b'(K)$ are the parameters of B and B' , respectively. In both cases, refer to Figure 23, V exhibits a clearly visible minimum. Its location depends upon the size of the information granules used in the model.

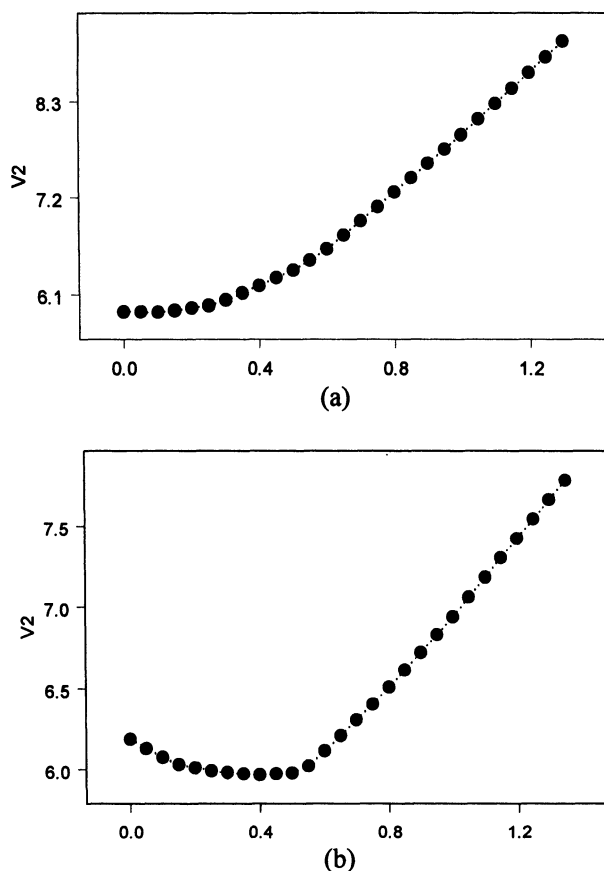


Figure 23. V versus T for two selected values of the granulation window equal to 8 (a) and 9 (b).

16. 6 ROUGH SETS IN SIGNAL GRANULATION

Rough sets arise as an interesting and appealing alternative in signal analysis. Consider that a signal is sampled with a fixed temporal window T while the granulation along the amplitude is completed by means of numeric intervals \mathbf{A} distributed uniformly across this variable (meaning that we exercise the standard digital quantization of the signal). Theoretically, a collection of these intervals forms an indiscernibility relation. The signal falling in the realm of the temporal window T , see Figure 24, gives rise to a certain interval that can be directly represented in the form of its lower and upper approximation. In essence, the result is some rough set $X = \langle X_-, X_+ \rangle$. Technically, both X_- and X_+ can be compactly represented as Boolean

vectors, e.g., $X_- = [1 \ 1 \ 0 \ 0 \ 1]$ and $X_+ = [1 \ 1 \ 1 \ 0 \ 1]$. The granular representation of the signal is an initial point for further modeling and signal classification.

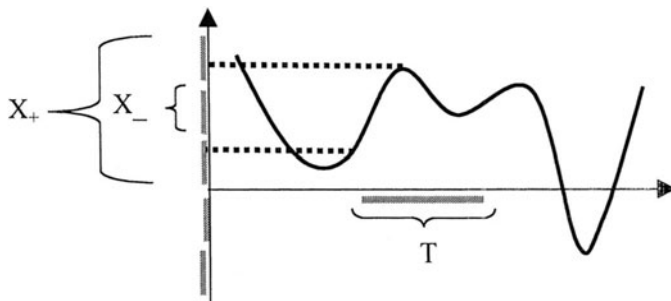


Figure 24. Rough set representation of a temporal signal.

For instance, we can construct rough set signal predictors in which knowing the current information granule in actual time moment, that is a rough set $X(k) = \langle X_-(k), X_+(k) \rangle$, one can predict rough set for “ $k+1$ ”-th instant. Schematically, we can portray a model of the form shown in Figure 25 where the mapping has been learnt e.g., in the structure of some neural network (NN). An interesting optimization task arises along the line of the design of the indiscernibility relation composed of nonuniformly distributed intervals in the amplitude space.

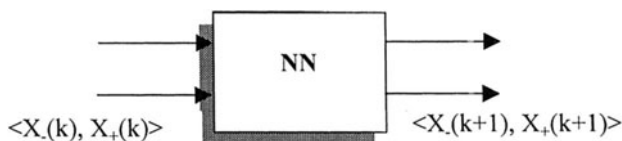


Figure 25. A neural network model for the prediction of granular signal realized via rough sets.

16. 7 CONCLUSIONS

In this chapter, we have discussed the role of information granulation and the ensuing information granules in description of time series. The detailed algorithm of information granulation produces information granules - fuzzy sets that are legitimate in terms of experimental data while still sustain their specificity. It has been shown that information granules can be regarded as generic conceptual entities contributing to the description of numeric time series. In this capacity, they are used as building blocks aimed at achieving high level, compact, and comprehensible models of signals. More importantly, the phase of information granulation could be viewed as a prerequisite to more synthetic and abstract processing such as the one encountered in syntactic pattern recognition.

REFERENCES

- Bargiela, A., Pedrycz, W. (2002), Recursive information granulation: Aggregation and interpretation issues, *IEEE Trans. on Syst. Man and Cybernetics*, to appear.
- Bargiela, A., Pedrycz, W. (2002), From numbers to information granules: A study in unsupervised learning and feature analysis, in: *Hybrid Methods in Pattern Recognition* (Bunke, H., Kandel, A. eds.), World Scientific, 75-112.
- Cios, K., Pedrycz, W., Swiniarski, R. (1998), *Data Mining Techniques*, Kluwer Academic Publishers, Boston.
- Horowitz, S.L. (1975), A syntactic algorithm for peak detection in waveforms with applications to cardiography, *CACM*, 18 (5), 281-285.
- Kundu, M., Nasipuri, M., Basu, D.K. (2000), Knowledge-based ECG interpretation: a critical review, *Pattern Recognition*, 33, 351-373.
- Papakonstantinou, G., Skordolakis, E., Gratazali, F. (1986), An attribute grammar for QRS detection, *Pattern Recognition*, 19(4), 297-303.
- Papakonstantinou, G. (1981), An interpreter of attribute grammars and its application to waveform analysis, *IEEE Trans. Software Engineering*, SE-7(3), 279-283.
- Skordolakis, E. (1986), Syntactic ECG pattern processing: a review, *Pattern Recognition*, 19(4), 305-313.
- Trahanias, P., E. Skordolakis, E. (1990), Syntactic pattern recognition of the ECG, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-12 (7), 648-657.
- Udupa, K., Murphy, I.S.N. (1980), Syntactic approach to ECG rhythm analysis, *IEEE Trans. on Biomedical Eng.*, BME-27(7).

GRANULAR DATA COMPRESSION

17. 1 INTRODUCTION

Images are examples of fuzzy relations. A pixel of the image assumes some numeric value that after normalization to the $[0,1]$ interval is an element of such fuzzy relation. Compression of fuzzy relations focuses on information granules and reduces irrelevant details. In this chapter, we discuss an approach to data compression carried out in terms of fuzzy relational equations. We reveal a way in which fuzzy relations and the computations therein naturally lend themselves into the way of information compression in image processing.

First (in Section 2) we elaborate on fuzzy relational equations. This brief summary captures the main ideas and provides a list of key solutions to the equations as well as discusses its character. In Section 3 we revisit the equations as a basic vehicle of image processing. Experimental studies are covered in Section 4. In the ensuing discussion fuzzy relations and images are treated as synonyms. Subsequently, we will be using the term fuzzy relation and image interchangeably. Referring to the previously mentioned papers, one should stress, however, that the approach studied here is fundamentally different from the concepts of image compression (or segmentation) presented in the literature.

17. 2 FUZZY RELATIONAL EQUATIONS: A BRIEF OVERVIEW

In what follows, we concentrate on the two well-known types of fuzzy relational equations quite frequently encountered in the literature. We recall main theoretical findings regarding solutions of these equations and analyze main properties of the equations (and their solutions) that are pertinent to this study. To avoid any unnecessary mathematical details, we confine ourselves to continuous t-norms (Butnariu and Klement, 1993) and cast the problem in finite universes of discourse A and B , $\text{card}(A), \text{card}(B) \leq \infty$. The latter assumption is in line with all applications in image processing where images are composed of a finite number of pixels.

A direct fuzzy relational equation, fuzzy relational equation, for short, assumes the following form

$$g = (A \times B) \bullet R \quad (1)$$

where A and B are two fuzzy sets defined in A and B , respectively, while R is a fuzzy relation defined in the Cartesian product of these spaces, that is $A \times B$, and $g \in [0,1]$. The choice of the introduced form of this equation will become obvious in the course of further discussion. Rewriting (1) in the form of the respective membership functions we obtain

$$g = \max_{\substack{x \in A \\ y \in B}} [A(x) \mathbf{t} B(y) \mathbf{t} R(x, y)]$$

The operation above is usually referred to as a max-t composition (convolution) of A , B , and R . For each combination of arguments (x and y), the convolution operation applies an *and* operation and returns a maximal value derived over the two universes of discourse. A useful interpretation of (1) allows us to treat " g " as a degree of possibility of R with respect to A and B .

The inverse problem calls for a solution of (1) for A , B and " g " provided and the fuzzy relation R regarded as unknown. More specifically, if (1) is solvable, there is a family of solutions (Di Nola et al., 1989). As being unique, the maximal solution is of particular interest. We recall the following fundamental result

- the solution set to (1) is nonempty iff (if and only if) the greatest solution (in sense of the inclusion relation) is expressed as

$$\hat{R} = (A \times B) \rightarrow g$$

or, equivalently

$$\hat{R}(x, y) = (A(x) \mathbf{t} B(y)) \rightarrow g$$

Here the symbol \rightarrow stands for the pseudocomplement (residuation) associated with the \mathbf{t} -norm standing in the original equation. The proof could be found e.g., in Pedrycz (1995).

Another important class of equations, referred to as adjoint fuzzy relational equations (Di Nola et al., 1989) assumes the form

$$g = (A \times B) \rightarrow R \quad (2)$$

with A , B , g , and R being already defined above. The membership value of g reads as

$$g = \min_{\substack{x \in A \\ y \in B}} [A(x) \text{t} B(y) \rightarrow R(x, y)]$$

The solution to (2) for A , B , and " g " available (assuming that the respective solution set is nonempty) reads as follows

$$\tilde{R}(x, y) = A(x) \text{t} B(y) \text{t} g$$

The fuzzy relation defined above gives the least element of the entire family of solutions (again viewed in the sense of inclusion of fuzzy relations).

In the sequel, the results are extended to the family of fuzzy relational equations - a situation that is far more important when dealing with various problems. Again, the theoretical findings are available considering that such equations are solvable. We are concerned with the systems of equations of the form

$$g_k = (A_k \times B_k) \bullet R \quad (3)$$

and

$$g_k = (A_k \times B_k) \rightarrow R \quad (4)$$

with $k=1, 2, \dots, N$. Again R is treated as an unknown fuzzy relation while the triples (A_k, B_k, g_k) $k=1, 2, \dots, N$ are provided. Then the maximal solution to (3) is an intersection of the respective maximal solutions to the successive equations occurring in the system under discussion,

$$\hat{R} = \bigcap_{k=1}^N (A_k \times B_k) \rightarrow R$$

The minimal solution to the system of equations (4) comes in the form

$$\tilde{R}(x, y) = \bigcup_{k=1}^N (A_k(x) \text{t} B_k(y) \text{t} g_k)$$

where the result is an union of the solutions to the individual equations.

17.3 RELATIONAL CALCULUS IN IMAGE COMPRESSION

Images are two-variable fuzzy relations. Let us now assume that we are provided with a collection of fuzzy sets - receptive fields or reference fuzzy sets defined over the x- and y - coordinate of the fuzzy relation (image). More specifically, let $\mathbf{A} = \{A_1, A_2, \dots, A_n\}$ and $\mathbf{B} = \{B_1, B_2, \dots, B_m\}$ denote the corresponding families of these fuzzy sets. Our intent is to express R in terms of the elements of \mathbf{A} and \mathbf{B} . As “n” and “m” is far lower than the dimension of the image itself, by doing that we end up with an evident effect of compression of the fuzzy relation (image) we have started with. The two composition operators discussed assume an transparent interpretation.

The max-t composition of A_i and B_j with the image matrix returns a number (a possibility degree)

$$g_{ij} = (A_i \times B_j) \bullet R$$

The elements g_{ij} collected together form a new and reduced (in comparison to the original image) fuzzy relation $G=[g_{ij}]$ that, in fact, is a compressed version of R (with the compression carried out with the aid of \mathbf{A} and \mathbf{B}). Evidently, different reference fuzzy sets yield different fuzzy relations G and deliver a different level of image compression. To make the discussed approach fully consistent with the way of designing compression mechanisms, we may refer to \mathbf{A} and \mathbf{B} as codebooks of linguistic terms utilized in the compression scheme. Here g_{ij} can be viewed as a final result of the logical convolution of the linguistic receptive field with the original content of the image; refer also to Figure 1.

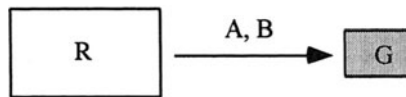


Fig.1. Compression of R with the use of the fuzzy codebook (A, B) .

The convolution operation itself is built as the max-t convolution of the elements of the codebook and the fuzzy relation with “t” being any continuous t-norm commonly exploited in the realm of fuzzy sets. Alluding to the semantics of the max-t composition itself, one may regard the result of the composition of A , B , and R as an effect of a two-phase processing, namely:

- logical *and* - type combination of A_i , B_j and R (completed via a certain t-norm). The operation is completed at the level of the individual pixels
- logical *or* - type summarization in the partial results produced through the OR operation

Altogether, one may regard the result (g_{ij}) as an optimistic aggregate of the content of the fuzzy relation of the image.

Interestingly, the resulting set - relation composition is highly nonlinear and can help focus on some specific portions of the image itself. The specific focal points are reached (attenuated) by distributing the elements of the codebook throughout the image.

Similarly, the collection of the adjoint fuzzy relation equations

$$g_{ij} = (A_i \times B_j) \rightarrow R$$

gives rise to another form of the compressed fuzzy relation R .

The reconstruction process uses the compressed version of R and the collection of the receptive fuzzy fields (that are fuzzy sets in the families **A** and **B**). Moreover, as the compression scheme has been already used, some specific convolution operations, the associated reconstruction (decompression) mechanism needs to be exploited in the setting of fuzzy relational equations. There are a number of questions regarding the efficiency of the compression - decompression method itself. Especially, we are interested in the way in which fuzzy landmarks (the elements of the codebooks) affect the performance of the compression, both in the sense of the obtained compression rate as well as the associated performance of the reconstruction itself.

Some useful findings stem directly from the theory of the relational equations and hinge on an analysis of the properties of the codebooks (fuzzy sets). Especially, we show that the granularity of the elements (fuzzy sets) of the codebook has a direct impact on the quality of the decompressed image (fuzzy relation). First, let us introduce a concise notation by considering a Cartesian product of A and B with these two being fuzzy sets coming from the respective codebooks, $\underline{A} = A \times B$. The notion of monotonicity exhibits an interesting semantics arising in the light of granularity and compression. Consider the compression-decompression scheme governed by (1).

If $\underline{A}' \subset \underline{A}$ then the decompressed image, say $\hat{R}(\underline{A}')$ is closer to R than $\hat{R}(\underline{A})$ is to R . More specifically, the following inequality holds

$$R \subset \hat{R}(\underline{A}') \subset \hat{R}(\underline{A})$$

This inclusion is easily quantified by the Hamming distance that leads to the inequality

$$\|\hat{R}(\underline{A}') - R\| \leq \|\hat{R}(\underline{A}) - R\|$$

In other words, the more *specific* \underline{A} is, the better the upper bound of the decompressed image. In some close sense (assuming that we are dealing with normal fuzzy sets), the specificity of \underline{A} deals with the granularity of the elements of \mathbf{A} . The higher number of the reference fuzzy sets in \mathbf{A} implies their higher granularity.

Two interesting boundary conditions imposed on the reference fuzzy sets are worth stressing. If the codebooks \mathbf{A} and \mathbf{B} are composed of singletons (viz. fuzzy sets with only a single element with membership equal one and all remaining values equal zero) and \mathbf{A} and \mathbf{B} are fuzzy partitions of the coordinates of R , then the achieved compression rate is 1 (no compression at all) and the reconstruction becomes perfect.

Subsequently, if \mathbf{A} and \mathbf{B} are composed of only one set defined over the entire space of x or y - coordinates, then the compression rate is the highest, however the quality of reconstruction is usually not acceptable. Note that under such circumstances all entries of G become the same and equal

$$g_{ij} = \max_{x,y} [A_i(x)tB_j(y)tR(x,y)] = \max_{x,y} R(x,y) = \text{hgt}(A)$$

where $\text{hgt}(\cdot)$ denotes the height of the fuzzy relation. In the sequel, the reconstruction formula produces the result

$$\hat{R}(x,y) = \max_{x,y} [A_i(x)tB_j(y)tR(x,y)] = \max_{x,y} R(x,y) = \text{hgt}(R)$$

If R is normal, then the reconstruction is nothing but a fuzzy relation with all entries set to 1 - this relation arises as a completely meaningless result of the completed reconstruction.

In general, the monotonicity property is satisfied meaning that elements of the codebook of higher granularity yield lower values of the reconstruction error. It is interesting to note that if \mathbf{A} and \mathbf{B} are composed of nonoverlapping sets (intervals) then the computations reveal some useful hints as to the distribution of the elements of the codebook. The calculations of the compressed fuzzy relation G are straightforward and very much simplified due to the form of the elements of the codebook. Note that the following holds

$$g_{ij} = \max_{x,y} [A_i(x)tB_j(y)tR(x,y)] = \max_{\substack{x \in \text{supp}(A_i), \\ y \in \text{supp}(B_j)}} R(x,y)$$

This reduces the computations of finding a maximum over the supports of the A_i and B_j . The reconstruction formula specifies as follows,

$$\hat{R}(x, y) = 1 \rightarrow g_{ij} = 1 \rightarrow \max_{\substack{x \in \text{supp}(A_i), \\ y \in \text{supp}(B_j)}} R(x, y)$$

for all (x, y) contained in the Cartesian products of the supports of A_i and B_j . The higher the variability over the supports of these two elements of the codebook, the more demanding becomes the problem of decompression of the R (and higher losses of information can be anticipated). This is mainly because of the effect of the absorption of the variability and its lower possibility of recovering out of the single entry of G (observe that the elements of the codebook do not overlap). It becomes evident from the expression (again this concerns only the elements in the abovestated region of R),

$$\varepsilon(x, y) = \max_{\substack{x \in \text{supp}(A_i), \\ y \in \text{supp}(B_j)}} R(x, y) - R(x, y)$$

that the reconstruction error is very much dependent on the size of the regions invoked by the respective elements of the codebook - we have already underlined the relationship of the reconstruction error by explicitly including A_i and B_j as the arguments in this error function.

The same monotonicity property can be derived for the adjoint fuzzy relational equations used in the compression - decompression mechanism.

Finally, Figure 2 illustrates several examples of fuzzy “points” - referential fuzzy sets starting from a genuine numerical quantity and progressing towards constructs of lower granularity.

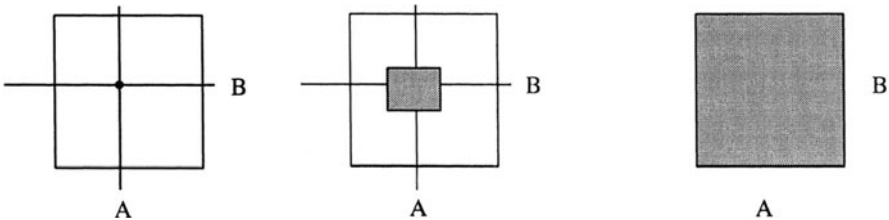


Figure 2. Examples of fuzzy points of different level of granularity.

One may think of another characteristics being central to the compression activities such as a compression ratio. In this case it could be taken as a ratio of the number of the original elements of the fuzzy relation R to the number of the compressed fuzzy relation G , $\text{card}(G)$, namely

$$\rho = \frac{\text{card}(G)}{\text{card}(\mathbf{A}) * \text{card}(\mathbf{B})}$$

Evidently, the lower the sizes of the codebooks, the higher the resulting compression rate. In the compression problem, the choice of the focal points - fuzzy sets is critical to the performance of the method. For illustrative purposes, we consider a one dimensional case (that could be then treated as a temporal signal) and specify a fuzzy set as a set (A_i).

The i -th coordinate of the fuzzy set G , g_i , is computed as

$$g_i = \bigvee_x [A_i(x) \wedge R(x)]$$

The reconstruction of R (assuming the use of the Godelian implication, that is a residuation implied by the minimum operation) is completed in the form

$$R(x) = A_i(x) \rightarrow g_i = \begin{cases} 1, & \text{if } A_i(x) \leq g_i \\ g_i, & \text{if } A_i(x) > g_i \end{cases}$$

Figure 3 illustrates how the reconstruction is carried out - note that the reconstructed membership function is constant over the window formed by the fuzzy sets used in this reconstruction. Note that we are interested in the reconstruction going on within the support of A_i . Apparently, when the variability of R goes up (or the size of A_i increases), the reconstruction quality deteriorates.

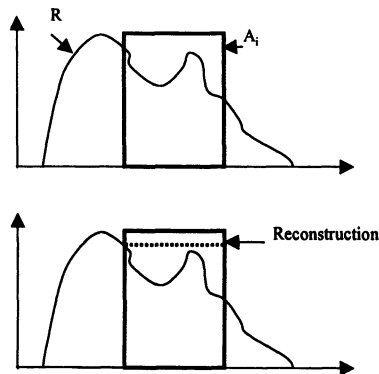


Figure 3. One dimensional reconstruction problem; see a description in the text.

This observation is a straightforward guideline as to the distribution of the focal points. In the next step, consider an image visualized in Figure 4. The distribution of

the levels of brightness is apparent as we have a collection of homogeneous vertical strips. Following the already presented arguments, it's enough to have only a single fuzzy set across the vertical coordinate and distribute a series of fuzzy sets horizontally and allocate them for an individual strip of the image.

In this way, the variability of the portion of the image encapsulated within each becomes reduced (these segments are highly homogeneous). Obviously, images are less regular than the one visualized here, yet the same rule of homogeneity does apply. Obviously, we may resort to more sophisticated distribution tools of the reference fuzzy sets such as evolutionary computing. The use of fuzzy clustering is also a viable choice.

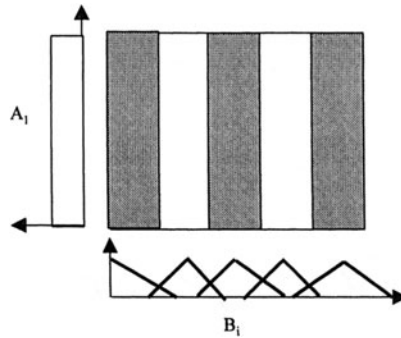


Figure 4. The distribution of the fuzzy sets reflected the variability of pixels existing in the corresponding region of the image.

17. 4 EXPERIMENTS

In this experimental part, we focus on the linguistic compression of Duke, a well-known mascot of JAVA. The original image is shown in Figure 5.

In all experiments we contrast the original image with its reconstruction. The quality of reconstruction of the decompressed image is expressed by the sum of absolute differences between these two images (the Hamming distance). We start with the direct fuzzy relational equations. The reference fuzzy sets were specified as triangular fuzzy numbers (case a) and those with Gaussian membership functions (case b of Figure 6). The plots of the membership functions of the reference fuzzy sets are included in Figure 6.

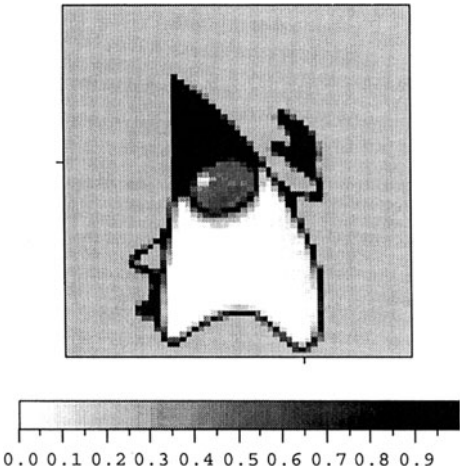


Figure 5. An original image of Duke.

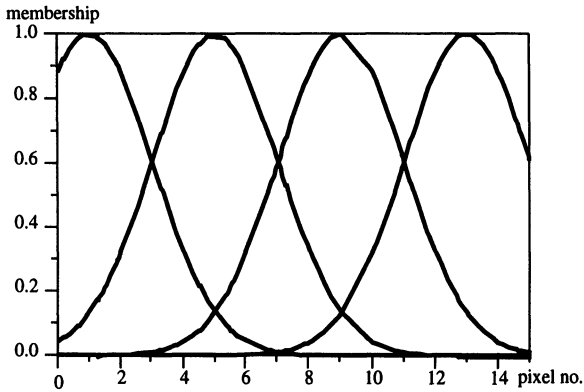
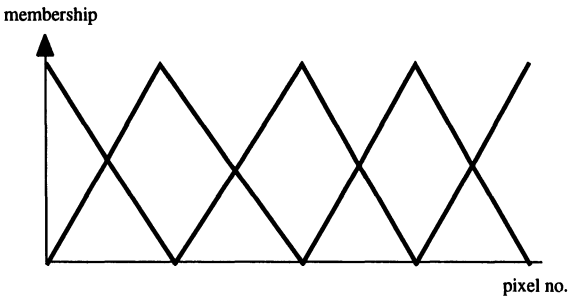


Figure 6. Examples of reference fuzzy sets (triangular and Gaussian membership functions).

We also vary the number of the reference fuzzy sets in order to gain a better insight into the compression rate and the quality of the reconstructed images. Furthermore we distribute the fuzzy sets uniformly throughout the image. Under such circumstances the changes in the number of the linguistic terms amount to a different size of the codebook supporting the compression procedure.

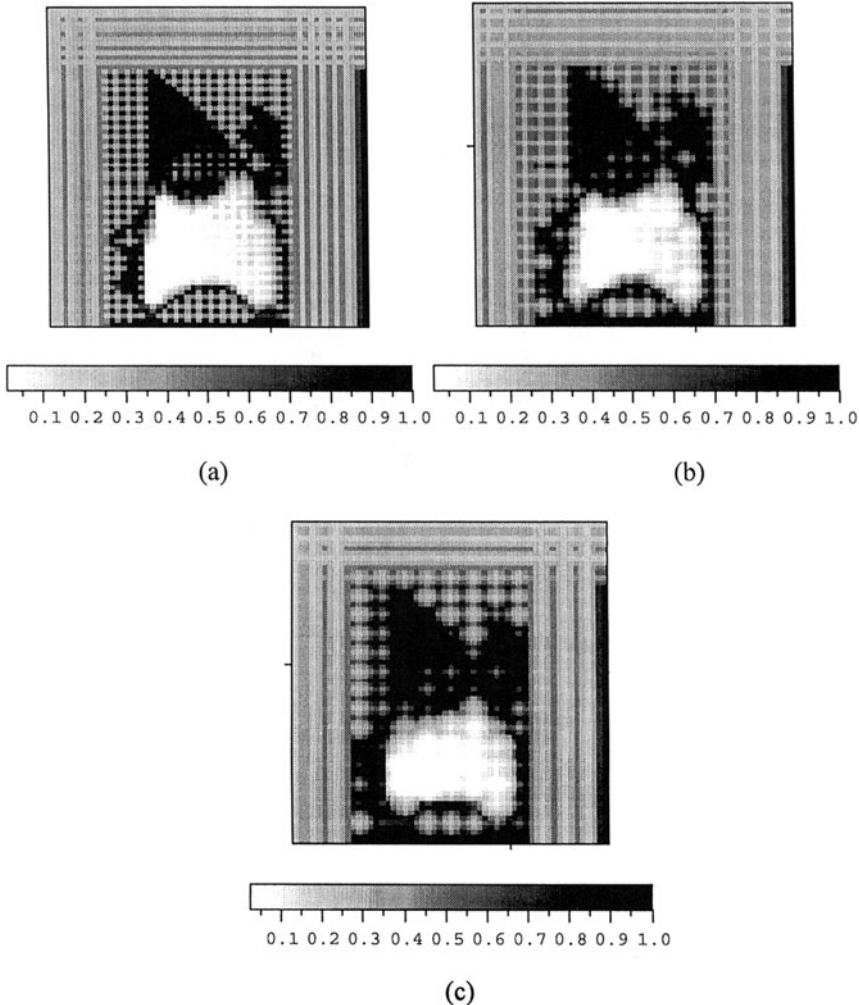


Figure 7. Decompressed image with the use of codebooks composed of triangular membership functions (t-norm: product) (a) fuzzy sets with modal values 2 pixels apart, (b) fuzzy sets with modal values 3 pixels apart, (c) fuzzy sets with modal values 4 pixels apart.

The series of figures, Figure 7, illustrates the performance of compression for the direct fuzzy relational equation. In all cases, the decompression produces a superset of the original fuzzy relation.

As expected, the reconstruction deteriorates when the size of the codebook increases. One can note an effect of some additional rastering showing up in the reconstructed image. It associates with the form of the membership functions that contribute to the brittleness phenomenon.

The quality of reconstruction can differ depending on the specific type of the t-norm used in the compression model. For illustrative purposes, Figure 8. illustrates the use of the minimum operation (and the resulting Godel implication). The minimum operation contributes to an even higher brittleness than before - note a quite visible raster coming with the body of the image.

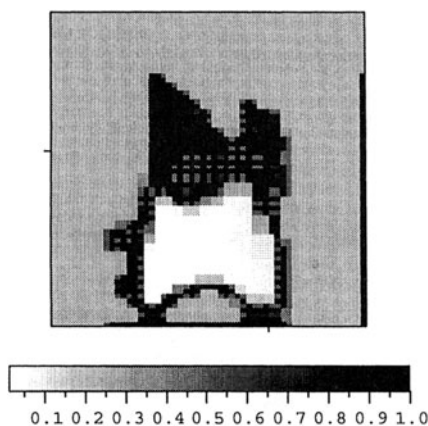


Figure 8. Decompressed image with the use of codebooks composed of triangular membership functions (t-norm: minimum) and fuzzy sets with modal values 2 pixels apart.

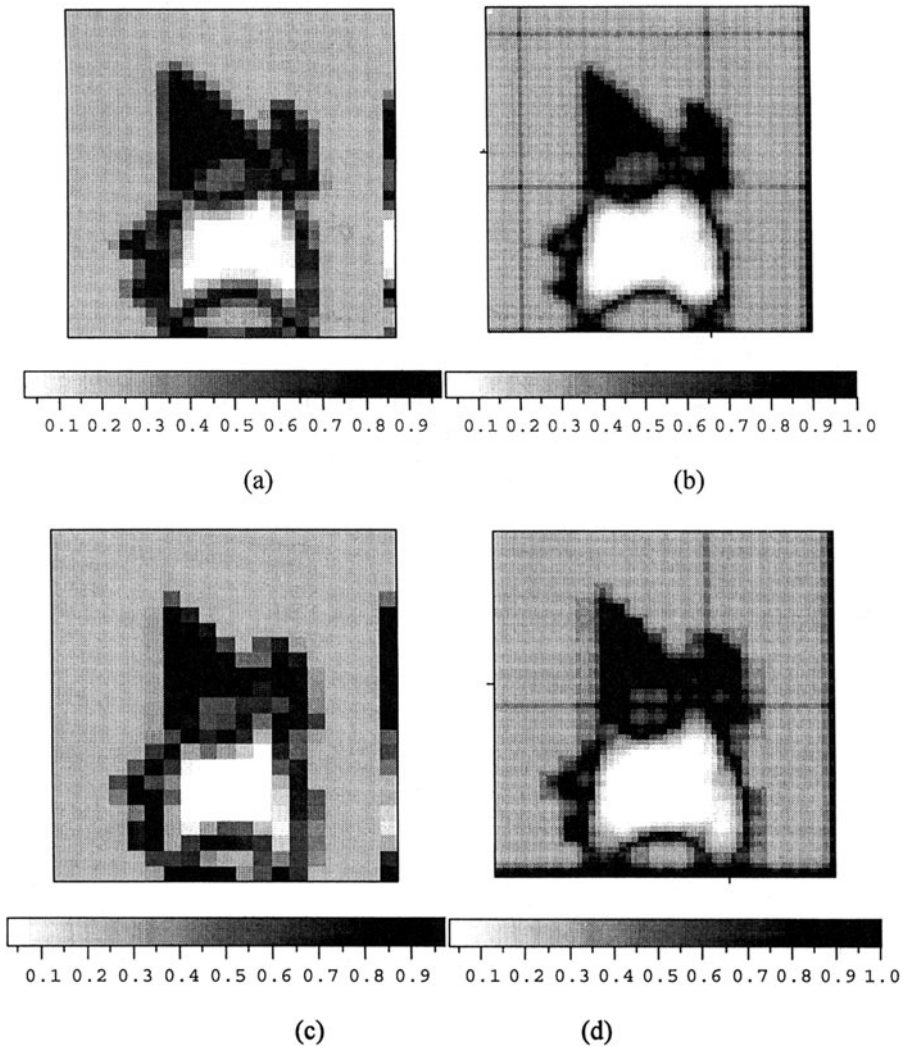
b. The codebook built with the use of the Gaussian membership functions. Here we specify them in the form

$$A(x) = \exp\left(-\frac{(x - x_0)^2}{\sigma}\right)$$

where σ stands for a spread of the reference fuzzy set while x_0 denotes a modal value which this fuzzy set is centered around. The use of the Gaussian membership

functions adds up to the computational overhead of the calculations yet it is of interest to investigate this alternative from the conceptual standpoint (taking into consideration a smooth nature of these membership functions).

The results organized as before, show the results of compression and decompression for $\sigma = 8$ and the modal values distant from each other by 2, 3, and 4 pixels, see Figure 9.



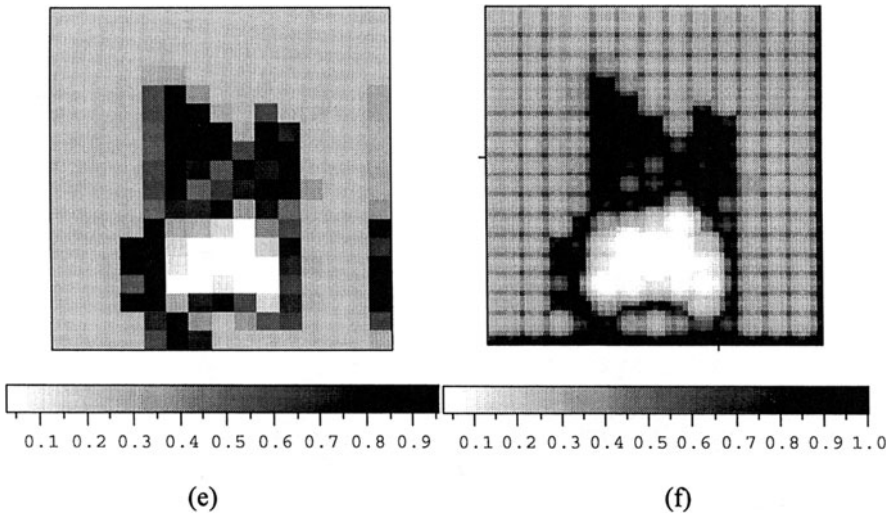


Figure 9. Compression with the use of codebooks composed of Gaussian membership functions (t-norm: product) and spreads equal 8: fuzzy sets with modal values 2 pixels apart (a) compressed fuzzy relation G, (b) decompressed image, fuzzy sets with modal values 3 pixels apart (c) compressed fuzzy relation G, (d) decompressed image, fuzzy sets with modal values 4 pixels apart (e) compressed fuzzy relation G, (f) decompressed image.

In comparison to the previous codebook, the rastering effect is less visible with an exception of a few strips in the background of the reconstructed image. With the decreasing size of the codebook, the quality of reconstruction deteriorates however in a different way than observed in the previous case - the image gets blurred. The size of the spread of the Gaussian function affects the quality of reconstruction. For comparison, Figure 10 shows the reconstructed image for $\sigma = 4$. Note more significant rastering effect that comes as a result of less substantial summarization effect as the fuzzy sets involved embrace less pixels.

The performance index expressing a Hamming distance between the image and its reconstructions for the triangular and Gaussian membership functions used in the compression is shown in Figure 11. It is quite evident that the Gaussian membership functions do a better job than the triangular membership functions of the previous codebook.

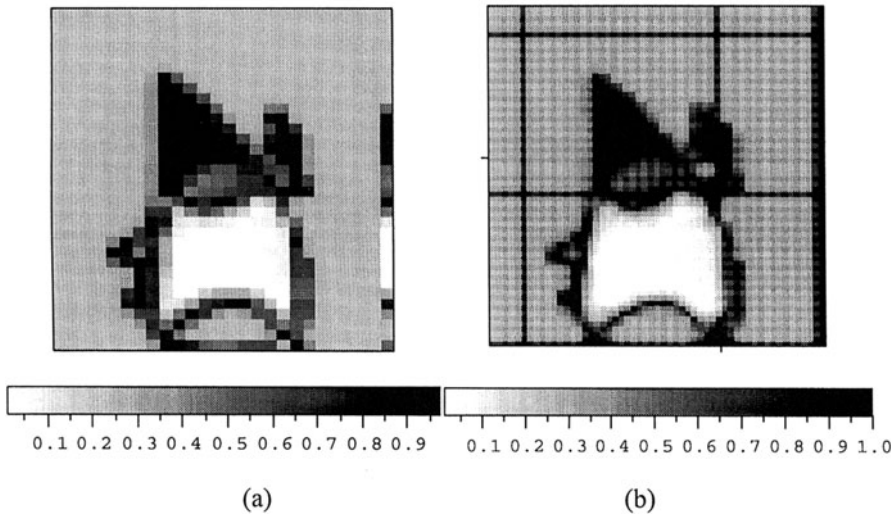


Figure 10. Compression with the use of codebooks composed of Gaussian membership functions (t-norm: product) and spreads equal 4; fuzzy sets with modal values 2 pixels apart; (a) compressed fuzzy relation (G), (b) decompression result.

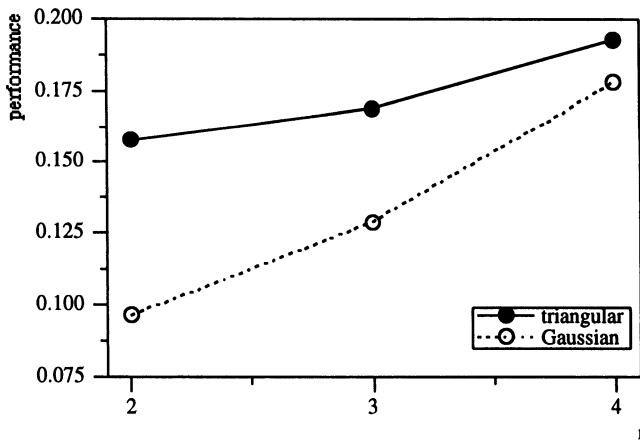


Fig.11. Performance index for compression using triangular and Gaussian fuzzy sets of the codebook.

We contrast these compression results with the standard way of compressing images by forming square windows of $p \times p$ pixels and averaging the levels of brightness occurring therein. The results, Figure 12, illustrate the successive compressions with

the windows with $p=2, 3$, and 4 . The effect of blocking the pixels is apparent; the image loses a lot of details and is barely comprehended. In the comparison with the use of the fuzzy relational equations, it becomes evident that they retain more useful details.

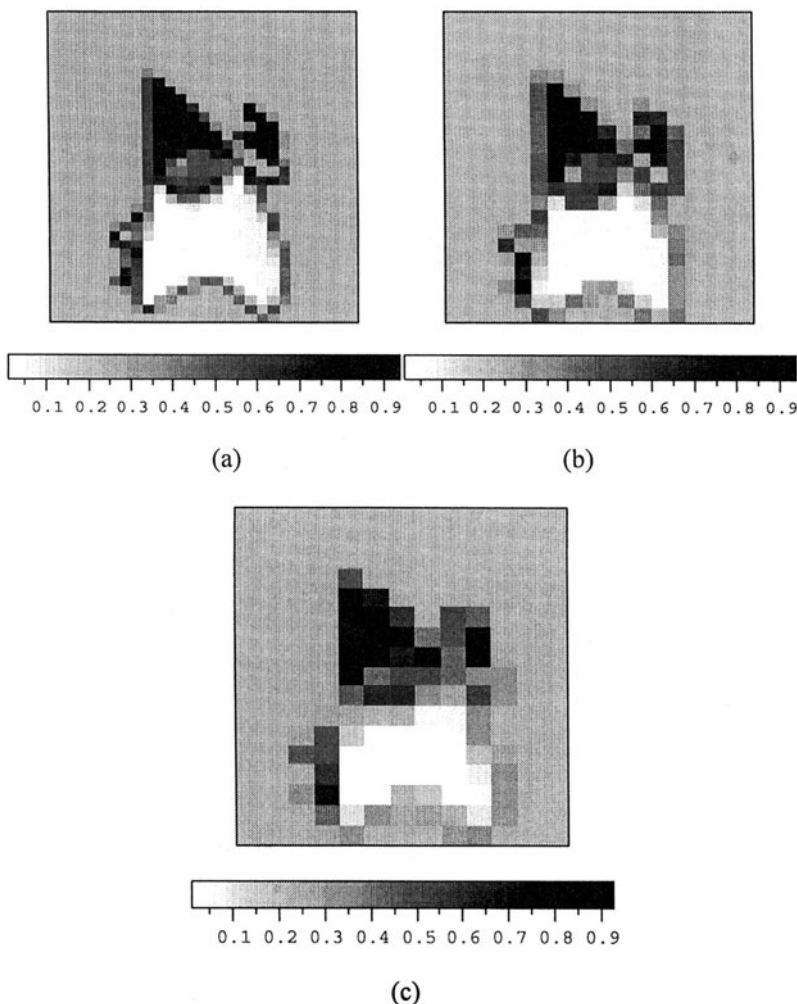


Figure 12. Compression through pixel blocking:(a) $p=2$, (b) $p=3$, (c) $p=4$.

Finally, we consider the adjoint fuzzy relational equations as a framework of fuzzy compression and decompression. As revealed in Section 2, the reconstructed fuzzy

relation is a lower bound of the original image. This phenomenon is very visible in Figure 13 where both triangular as well as Gaussian codebook were considered.

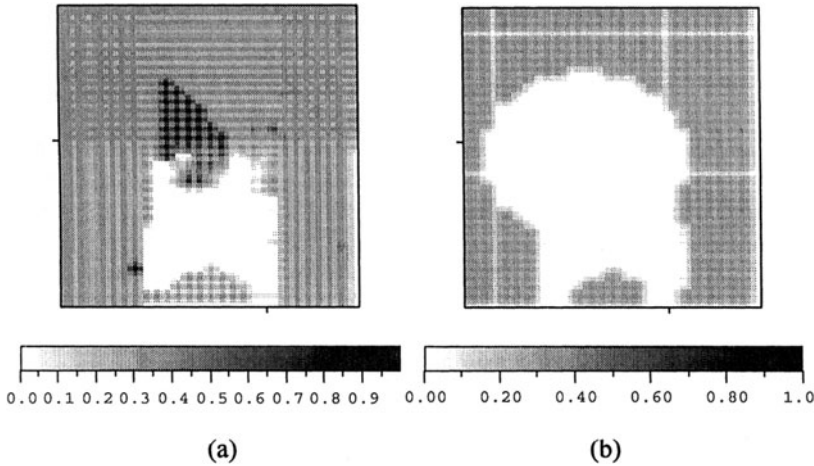


Figure 13. Compression with the use of adjoint fuzzy relational equation (t-norm: product) and triangular (a) and Gaussian (b) membership functions.

It is also visible that the reconstruction comes with a far higher reconstruction error leading to a fairly ghost-like image of Duke. This poor reconstruction can be explained by the specific instance of the t-norm and the nature of the image itself.

17.5 CONCLUSIONS

The study has introduced a concept of fuzzy relation calculus to the problems of image processing. We have discussed fuzzy relation - based data compression. The methodology of data compression hinges on the theory of fuzzy relational equations where the solutions to the specific class of equations give rise to a reconstructed fuzzy relation (image). The main properties of fuzzy relational equations were introduced and analyzed with respect to the resulting reconstruction and compression capabilities. We showed that the compression and decompression (reconstruction) abilities rely also on the granularity of the referential fuzzy sets used to compress the image. Owing to the diversity of t- norms occurring in these fuzzy relational equations, we can think of them as an additional vehicle of parametric flexibility supporting the development of the compression models. We analyzed and illustrated the nature of the boundary conditions of the fuzzy relational equations. It is worth emphasizing that the reconstruction yields either a lower or upper bound of the original image. The upper bound of the image constructed via the relational calculus could be of particular interest especially in cases when we are

concerned far more about missing details rather than overestimating their size. It should be stressed that the method does not employ any preprocessing leading to the codebooks of higher efficiency (the codebooks are quite often formed by various clustering mechanisms as discussed e.g., by Karayiannis and Pai (1995). While interesting and definitely more promising, this preprocessing phase accounts for a relatively significant computational overhead. This overhead is, however, avoided in this study. Obviously, one may anticipate the exploitation of the clustering techniques upfront and the utilization of such findings (that is prototypes and membership functions) in the completion of the relational construct - this will be of interest to pursue in the future. One should stress that the study undertaken in this study is quite distinct from the existing compression methods (Salomon, 1998; Tsay et al., 1996; Wu, 1996; Wu and Sung, 1994) in the sense they explicitly exploit application-oriented information granules (that is reference fuzzy sets).

The proposed approach sheds light on the vital links between fuzzy sets and fuzzy relational equations, in particular and image processing - a promising research area worth further investigations. The challenging and promising direction would be to concentrate on compression mechanisms dealing with heterogeneous information associated inherently with any advanced Internet endeavors. In this context, one should emphasize a predominant role of granular computing; here information granules are building blocks of the compression mechanism.

REFERENCES

- Butnariu, D., Klement, E.P. (1993), *Triangular Norm-based Measures and Games with Fuzzy Coalitions*, Kluwer Academic Publishers, Dordrecht.
- Di Nola, D., Sessa, S., Pedrycz, W., Sanchez, E. (1989), *Fuzzy Relation Equations and Their Applications to Knowledge Engineering*, Kluwer Academic Publishers, Dordrecht.
- Karayiannis, N.B., Pai, P.I. (1995), Fuzzy vector quantization algorithms and their application in image compression, *IEEE Transactions on Image Processing*, 4, 1193- 1201.
- Pedrycz, W. (1995), *Fuzzy Sets Engineering*, CRC Press, Boca Raton, FL.
- Salomon, D. (1998), *Data Compression. The Complete Reference*, Springer-Verlag, New York.
- Tsay, M.K., Huang, J.F., Chung, W.P. (1996), Image compression using VQ and fuzzy classified algorithm, *Proc. IEEE Int. Conference on System, Man and Cybernetics*, Beijing, October 14-17, 1996, vol. I, 466 - 471.
- Wu, C.J. (1996), Performance comparison of fuzzy and proportional controllers in the application of image compression, *Proc. IEEE Int. Conference on System, Man and Cybernetics*, Beijing, October 14-17, 1996, vol. I, 378 - 383.
- Wu, C.J., Sung, A.H. (1994), The application of fuzzy controller to JPEG, *Electronics Letters*, 30, 1375-1376.

INTERVAL STATE ESTIMATION IN SYSTEMS MODELLING

This chapter focuses on interval information granules that arise in the context of state estimation of systems that are monitored with limited accuracy. For these systems, the representation of state uncertainty as confidence intervals offers significant advantages over the more traditional approaches with probabilistic representation of noise. While the filtered-white-Gaussian noise model can be defined on grounds of mathematical convenience, its use is necessarily coupled with a hope that an estimator with good properties in idealized noise will still perform well in real noise. In this study we propose a more realistic approach of matching the noise representation to the extent of prior knowledge. Both interval and ellipsoidal representation of noise illustrate the principle of keeping the noise model simple while allowing for iterative refinement of the noise as we proceed. We evaluate one non-linear and three linear state estimation technique both in terms of computational efficiency and the cardinality of the state uncertainty sets. The techniques are illustrated on a synthetic and a real-life system.

18. 1 INTRODUCTION

Estimation of a state of a system that is monitored through measurements that have limited accuracy has long been recognized as a challenging practical problem. This is primarily because we are no longer interested in just a single numerical value but try to identify a much larger set of all possible system states. If the measurement errors are described well by some probability distribution functions then the set of feasible state estimates can also be described by a probability distribution function. Unfortunately, in most practical situations, this is not possible as one deals with observations that do not have full statistical characterization. An approach to dealing with such situations has been introduced by Schweppe (1973) and further developed by other researchers (Fogel, 1982; Bargiela, 1989; Hainsworth, 1988; Norton, 1986; Mo, 1988; Jaulin, 1996; Milanese, 1996; Kurzhanski, 1997). This is referred to in the literature as an unknown but bounded error approach.

Within this approach there is a full spectrum of methods that differ significantly with regard to computational complexity and the quality of the estimation of the uncertainty set (an extent to which this set is over estimated). An example of an accurate but computationally inefficient method is a Monte Carlo estimation that is guaranteed to avoid any over estimation but it requires a very large (in theory an infinite) number of iterations to cover the continuous (infinite) solution space. At the other end of the spectrum there is a trivial solution that takes the whole system space as an estimate of the system uncertainty state. This has a zero numerical complexity but also has no practical value. In between these extremes there are methods that try to achieve the best balance between computational complexity and accuracy. Among the better known are methods based on variants of the linear programming (Belforte, 1984; Bargiela, 1985; Norton, 1986) and the ellipsoidal bounding techniques (Fogel, 1982; Belforte, 1985; Kurzhanski, 1997). The main problem associated with the linear programming methods is their numerical complexity, particularly when one deals with multi-dimensional and highly constrained problems, (Norton, 1986). On the other hand, the ellipsoidal bounding techniques are computationally efficient but often provide loose approximation of the state uncertainty set. Furthermore the results of ellipsoidal bounding are dependent on the order of processing of constraints (Milanese, 1996).

In an effort to combine the advantages of the two approaches researchers have attempted reduction of the number of constraints using ellipsoidal bounding techniques followed by a linear programming estimation of the state uncertainty set that takes advantage of a smaller number of active constraints that need to be considered (Belforte, 1985; Arruda, 1991). Here we examine an alternative combination of techniques. Our proposed method takes its roots in the linear programming formulation but it avoids computationally expensive Simplex-type basis exchanges by taking advantage of the sensitivity matrix that is calculated during point estimation of the system state for average values of measurements. In other words, we hybridize the point estimation of the average system state with linear programming estimation of the state uncertainty set.

This chapter is organized as follows. In Section 2 we provide an overview of the four methods for estimation of the state uncertainty set. These are the Monte Carlo, linear programming, sensitivity analysis and ellipsoidal bounding methods. For the sensitivity analysis method we look at two alternative formulations. Each method is discussed in a separate subsection and is illustrated with a numerical example. Section 3 provides a real-life example of estimation of the state uncertainty set. An exactly determined- and an over-determined set of measurements are considered and hyperbox enclosures of the state uncertainty set are compared for all the state estimators concerned. Conclusions about the relative merits of the four state estimation techniques are provided in Section 4.

18.2 ESTIMATION OF THE STATE UNCERTAINTY SET

The estimation of a state of a system described by a (known) non-linear function $g(\cdot)$ and monitored through measurements that have a limited accuracy, is a process of finding a set of feasible state variables \mathbf{x} that results in values $g(\mathbf{x})$ that satisfy the measurement constraints.

To put it formally we identify a *state uncertainty set* $X(\cdot)$

$$X(M, \mathbf{z}^-, \mathbf{z}^+) := \{ \mathbf{x} \in \mathbf{R}^n : g(\mathbf{x}) \in Z(M, \mathbf{z}^-, \mathbf{z}^+) \} \quad (1)$$

where $Z(\cdot)$ represents a set of feasible measurement vectors, i.e.

$$Z(M, \mathbf{z}^-, \mathbf{z}^+) := \{ \mathbf{z}^0 \in \mathbf{R}^m : z_i^- \leq z_i \leq z_i^+, i=1, \dots, m \} \quad (2)$$

In the above, the \mathbf{z}^- and \mathbf{z}^+ represent vectors of lower and upper bounds on the measurements and \mathbf{z}^0 represents a specific instance of the measurement vector. The set $Z(M, \mathbf{z}^-, \mathbf{z}^+)$ represents an m -dimensional interval vector $[\mathbf{z}] = [\mathbf{z}^-, \mathbf{z}^+]$. Using the interval notation we can denote the state uncertainty set as $X([\mathbf{z}])$. The inclusion relationship in (1) is referred to in the literature as the *uncertain system equation*. For the uncertain system equation there is no unique operating state that can be calculated. All that can be defined is a set of possible operating states resulting from the set of possible measurement vectors. No preference is placed on these, all are assumed to be equally likely. Although the lack of a unique estimate of \mathbf{x} is at first worrying it reflects the reality that any physical system can be known only with a limited accuracy. So, the unknown-but-bounded (Schweppe, 1973) modelling of measurement uncertainty can be seen as a natural description of physical reality.

By analogy to the measurement uncertainty bounds it is natural to extend this approach to defining the uncertainty bounds on system state variables. These can be formalized as follows

$$\begin{aligned} x_i^- &:= \min_{\mathbf{x} \in X([\mathbf{z}])} x_i, \quad i=1, \dots, n \\ x_i^+ &:= \max_{\mathbf{x} \in X([\mathbf{z}])} x_i, \quad i=1, \dots, n \end{aligned} \quad (3)$$

The vectors \mathbf{x}^- and \mathbf{x}^+ provide lower and upper bounds on the state vector \mathbf{x} in the same way that \mathbf{z}^- and \mathbf{z}^+ did for the measurement vector. For each individual variable, the interval $[x_i^-, x_i^+]$ is referred to as the *uncertainty interval* for the i^{th} variable and x_i^- and x_i^+ are referred to as *confidence limits* (Bargiela, 1989; Hainsworth, 1988; Hartley, 1997). The uncertainty intervals or confidence limits, as defined in (3), are as tight as can be achieved with the given measurement

uncertainty. However, calculating these bounds is a challenging task and, depending on the simplifying assumptions, it can result in calculated bounds that are looser or tighter.

As a general point, it is worth noting that the hyperbox (interval vector $[x]$) defined by the uncertainty intervals $[x_i]$, $i=1, \dots, n$

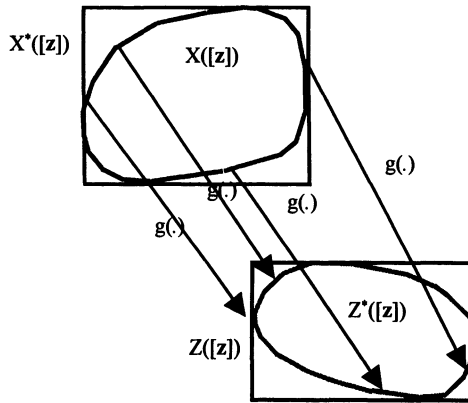
$$[x] = X^*([z]) := \{ x \in \mathbf{R}^n : x_i^- \leq x_i \leq x_i^+, i=1, \dots, n \} \quad (4)$$

is a superset of $X([z])$, i.e. $X([z]) \subseteq X^*([z])$. The set $X^*(.)$ contains in addition to $X(.)$ all those combinations of values of x_i that are each feasible for individual state variables but are not feasible for a state vector x .

Similarly, we notice that the set $Z^*(.)$, which is the image of the state uncertainty set $X(.)$ obtained through mapping $g(.)$ and formally defined as

$$Z^*([z]) := \{ z \in \mathbf{R}^m : z = g(x), x \in X([z]) \} \quad (5)$$

is a subset of the feasible measurement set $Z(.)$. The set $Z(.)$ typically contains, in addition to $Z^*(.)$, such z for which there is no x (neither in $X([z])$ nor \mathbf{R}^n) for which $g(x)=z$. In other words, there may be vectors in $Z([z])$ that are inconsistent for $g(.)$. These two remarks are illustrated in Figure 1 for a 2-dimensional case.



- $X([z])$ – The state uncertainty set
- $X^*([z])$ – The smallest box containing the state uncertainty
- $Z([z])$ – The measurement uncertainty set
- $Z^*([z])$ – The image of the state uncertainty set when mapped by $g(.)$

Figure 1. Relationship between X^* and X , and between Z^* and Z .

The optimization problems implied by (3) are, in general, quite complex, particularly if the function $g(\cdot)$ is non-linear. With n state variables and m measurements the confidence limit analysis requires $2n$ non-linear optimizations each subject to $2m$ constraints (the $2m$ constraints arise by considering the lower and upper bounds on the measurement uncertainty). For real-life systems there may be several hundred state variables and several hundred measurements. Therefore, confidence limit analysis is a highly computationally intensive task. We shall now focus on the assessment of the relative merits of the alternative confidence limit analysis algorithms.

Monte Carlo Method

In normal use, deterministic state estimators produce one state estimate for one measurement vector. Used in this way they give no indication of how a state estimate may vary in response to variations in the measurement values. However, if a deterministic state estimator is used repeatedly for a whole range of measurement vectors then some indication of state estimate variability is provided. It is this idea that forms the basis of the Monte Carlo approach to confidence limit analysis. A large number of feasible state estimates are generated, as randomly as possible, and from these the state estimate confidence limits are estimated. The larger the number of random feasible state estimates the more reliable the estimate of the confidence limits.

Let z^j be a measurement vector, selected randomly from the set $Z([z])$ and let x^j be a state estimate calculated using z^j . A state estimate x^j is a feasible if $g(x^j) \in Z([z])$. This follows from the definition of feasible vectors given in equation (1). However, for some z^j there is no state vector x for which $g(x) = z^j$ (see Figure 1). In fact, if $Z^*([z])$ is defined as in (5), then $Z^*([z]) = Z([z])$ only when M is a minimal measurement set (i.e. if M is an observable set and has no observable subset). For a sequence, z^1, \dots, z^k , of measurement vectors selected randomly from $Z([z])$, a sequence of sets X^1, \dots, X^k can be defined,

$$X^j := \{ x^j \in \mathbb{R}^n : g(x^j) \in Z([z]) \text{ for some } i \in \{1, \dots, j\} \}, j=1, \dots, k \quad (6)$$

where x^j is the state estimate calculated using z^j . This sequence of sets is such that $X^j \subseteq X^k \subseteq X([z])$ for all $j=1, \dots, k-1$, as only feasible state estimates are contained in X^j . For a large number, k , of randomly selected measurement vectors it can be assumed that X^k is approximately equal to $X([z])$. In other words, as $k \rightarrow \infty$, $X^k \rightarrow X([z])$.

The actual estimator used to calculate x^j of no importance, provided that it is unbiased and that it can guarantee convergence in a high proportion of cases. All

state estimates are checked for feasibility before being used to update X^j . A sequence of random measurement vectors can be selected from $Z([z])$ by using a random number generator. For example, a sequence of random numbers, r_1^j, \dots, r_m^j , scaled to be between 0.0 and 1.0, can be generated and used to construct the measurement vector z^j

$$z_i^j = z_i^- + r_i^j \cdot (z_i^+ - z_i^-), i=1, \dots, m \quad (7)$$

where z^- and z^+ are the lower and upper bounds for $Z([z])$. z^{j+1} can be constructed in a similar way from a new sequence of random numbers. Throughout the computations, for X^j , only two vectors need to be stored, these are x^- and x^+ , the lower and upper bounding vectors for the current set of feasible state estimates, X^j . These vectors are updated whenever a new feasible vector, not contained in any of the X^j 's, is found.

The Monte Carlo confidence limit algorithm

1. Select a large number, k (to limit the number of iterations) and set $i = 0$.
2. Set $i = i + 1$.
3. Select a sequence of m random numbers, r_1^i, \dots, r_m^i , and use these to construct a random measurement vector z^i from $Z([z])$ as indicated in (7).
4. Calculate a state estimate, x^i from z^i . If $g(x^i) \in Z([z])$, then use x^i to update x^- , x^+ . Otherwise reject x^i as infeasible.
5. If $i < k$, go back to step 2. Otherwise stop.

Monte Carlo method is obviously very demanding computationally, but despite this it is useful in some situations. The condition that only feasible state estimates are used to update x^- and x^+ makes the procedure mathematically reliable and ensures that these bounds can be attained. The method can be used as a yardstick, against which the accuracy of all other confidence limit algorithms can be compared. Unfortunately, the method is impractical in many real-time applications. More practical methods are described in subsequent sections.

Linear Programming Method

The application of the linear programming technique to the estimation of the state uncertainty set $X(\cdot)$ relies on the liberalization of the system function, $g(\cdot)$. This can be accomplished using a first order Taylor approximation to give

$$g(x) \approx g(\hat{x}) + J \cdot (x - \hat{x}) \quad (8)$$

for all state vectors close to \hat{x} . In (8), J is the Jacobian matrix evaluated at \hat{x} . Although $g(\cdot)$ can be linearized around any state vector, \hat{x} , it is best if \hat{x} is in some way central to the state uncertainty set. This is because, the approximation used in (8) is more accurate for values of x for which $\|x - \hat{x}\|$ is small. The best estimate

available for the center of $X([z])$ is the state estimate calculated from z^0 . So, the definition (1) can be linearized using (8) to give a linear approximation, $X^1([z])$, of the state uncertainty set $X([z])$. This is defined as follows:

$$X^1([z]) := \{ \mathbf{x} \in \mathbf{R}^n : \mathbf{g}(\hat{\mathbf{x}}) + \mathbf{J} \cdot (\mathbf{x} - \hat{\mathbf{x}}) \in Z([z]) \} \quad (9)$$

$X^1([z])$ will be referred to as the linearized state uncertainty set. Substituting \mathbf{dx} for the difference between the state vector and its estimate, $\mathbf{x} - \hat{\mathbf{x}}$, and using the definition of $Z([z])$ given in (2), $X^1([z])$ can be written as:

$$X^1([z]) := \{ \mathbf{x} \in \mathbf{R}^n : \mathbf{x} = \hat{\mathbf{x}} + \mathbf{dx}, \mathbf{z}^- - \mathbf{g}(\hat{\mathbf{x}}) \leq \mathbf{J} \cdot \mathbf{dx} \leq \mathbf{z}^+ - \mathbf{g}(\hat{\mathbf{x}}) \} \quad (10)$$

The set $X^1([z])$ has, in general, quite a complex topology and will not be calculated explicitly. Instead, the smallest hyperbox enclosure of $X^1([z])$ is sought. This set will be denoted by $X^{1*}([z])$ and referred to as the *linearized state uncertainty box*. Following the definition of \mathbf{x}^- and \mathbf{x}^+ in (3), lower and upper limits for $X^1([z])$ can be defined as follows:

$$\begin{aligned} x_i^- &:= \min_{\mathbf{x} \in X^1([z])} x_i, & i=1, \dots, n \\ x_i^+ &:= \max_{\mathbf{x} \in X^1([z])} x_i, & i=1, \dots, n \end{aligned} \quad (11)$$

Using the interval vector notation we can express the set $X^{1*}([z])$ as $[\mathbf{x}^1]$. The task of calculating the bounding vectors of $X^1([z])$ can now be formulated as a linear programming problem. To simplify this we introduce some auxiliary notation

$$\mathbf{dz}^- := \mathbf{z}^- - \mathbf{g}(\hat{\mathbf{x}}) \quad (12)$$

$$\mathbf{dz}^+ := \mathbf{z}^+ - \mathbf{g}(\hat{\mathbf{x}}) \quad (13)$$

$$DZ([z]) := \{ \mathbf{dz} \in \mathbf{R}^m : \mathbf{g}(\hat{\mathbf{x}}) + \mathbf{dz} \in Z([z]) \} \quad (14)$$

$$DX^1([z]) := \{ \mathbf{dx} \in \mathbf{R}^n : \hat{\mathbf{x}} + \mathbf{dx} \in X^1([z]) \} \quad (15)$$

$DX^1([z])$ is the set $X^1([z])$ shifted by $\hat{\mathbf{x}}$, $DZ([z])$ is the measurement uncertainty set shifted by $\mathbf{g}(\hat{\mathbf{x}})$ and \mathbf{dz}^- and \mathbf{dz}^+ represent 'tightest' lower and upper bounds for the set $DX^1([z])$. Then, the i^{th} element of \mathbf{dx}^- can be found by solving the linear programming problem

$$\begin{aligned} &\text{minimize } dx_i \\ &\text{subject to } \mathbf{dz}^- \leq \mathbf{J} \cdot \mathbf{dx} \leq \mathbf{dz}^+ \end{aligned} \quad (16)$$

Similarly, the i^{th} element of \mathbf{dx}^+ can be found by solving the corresponding linear programming problem

$$\begin{aligned} & \text{maximize } dx_i \\ & \text{subject to } \mathbf{dz}^- \leq J \cdot \mathbf{dx} \leq \mathbf{dz}^+ \end{aligned} \quad (17)$$

Hence by performing $2n$ linear programs, the vectors \mathbf{dx}^- and \mathbf{dx}^+ can be constructed. Once \mathbf{dx}^- and \mathbf{dx}^+ have been calculated, it is a simple matter to construct the bounds

$$\mathbf{x}^{1-} = \hat{\mathbf{x}} + \mathbf{dx}^- \quad (18)$$

$$\mathbf{x}^{1+} = \hat{\mathbf{x}} + \mathbf{dx}^+ \quad (19)$$

The analysis of the optimization problems (16) and (17) leads to the conclusion that the maximum discrepancy between the upper limits for any of the state variables in $X([z])$ and $X^1([z])$ is of order $O(\|\mathbf{x} - \hat{\mathbf{x}}\|^2)$, (Bargiela, 2001). Unfortunately, the solution of $2n$ linear programs with $2m$ constraints given by (16) and (17) represents a large computational task. This is particularly so if the solution is attempted by a direct application of the revised simplex, or any similar linear programming algorithm. In an attempt to alleviate this load an alternative formulation has been proposed in (Bargiela, 1989; Hainsworth, 1989). This is based on the partitioning of the Jacobian matrix onto a square, observable matrix and the remainder representing redundant measurements. Using this partitioning the optimization tasks (16) and (17) can be written as

$$\begin{aligned} & \text{minimize} && \mathbf{a}^i \cdot \mathbf{dz}^n \\ & \text{subject to} && (\mathbf{dz}^n)^- \leq \mathbf{dz}^n \leq (\mathbf{dz}^n)^+ \\ & && (\mathbf{dz}^{m-n})^- \leq J^{m-n} (J^n)^{-1} \mathbf{dz}^n \leq (\mathbf{dz}^{m-n})^+ \end{aligned} \quad (20)$$

and

$$\begin{aligned} & \text{maximize} && \mathbf{a}^i \cdot \mathbf{dz}^n \\ & \text{subject to} && (\mathbf{dz}^n)^- \leq \mathbf{dz}^n \leq (\mathbf{dz}^n)^+ \\ & && (\mathbf{dz}^{m-n})^- \leq J^{m-n} (J^n)^{-1} \mathbf{dz}^n \leq (\mathbf{dz}^{m-n})^+ \end{aligned} \quad (21)$$

where \mathbf{a}^i is the i^{th} row of $(J^n)^{-1}$, J^n is a square sub-matrix of J , J^{m-n} represents the portion of J corresponding to redundant measurements and \mathbf{dz}^n and \mathbf{dz}^{m-n} represent the measurement vectors associated with J^n and J^{m-n} .

This formulation of the problem has an important advantage over the formulation of (16), (17) since the number of constraints to consider is reduced to $2(m-n)$. In many real-life systems measurement redundancy is low so $m-n \ll m$. A disadvantage of the second formulation is that it requires the inversion of the matrix J^n . However J^n need only to be inverted once while the maximizations and minimizations are carried out $2n$ times. So, with an efficient matrix inversion scheme this disadvantage quickly disappears.

The linear programming confidence limit algorithm

1. Select an observable subset of M containing n measurements. This is the minimal measurement set and is denoted by M' . Order M with the elements of M' appearing first.
2. Re-order \mathbf{dz}^- and \mathbf{dz}^+ according to the new ordering of M . Assemble $(\mathbf{dz}^-)^-$, $(\mathbf{dz}^-)^+$, $(\mathbf{dz}^{m-n})^-$, $(\mathbf{dz}^{m-n})^+$, \mathcal{J}^n and \mathcal{J}^{m-n} .
3. Factorize \mathcal{J}^n and calculate $\mathcal{J}^{m-n}(\mathcal{J}^n)^{-1}$.
4. For each variable, $i=1, \dots, n$, calculate \mathbf{a}^i , the i^{th} row of $(\mathcal{J}^n)^{-1}$ and carry out the maximization in (20) using a linear programming method. The resultant value of $\mathbf{a}^i \cdot \mathbf{dz}^-$ is the i^{th} element of \mathbf{dx}^+ . Similarly, carry out the minimization in (21), to obtain the i^{th} element of \mathbf{dx}^- .
5. Add \mathbf{dx}^- and \mathbf{dx}^+ to $\hat{\mathbf{x}}$ to obtain \mathbf{dx}^{1-} and \mathbf{dx}^{1+} .

The computational complexity of (20), (21) is $O(n^2(m-n))$ compared to $O(n^2m)$ of optimizations (16), (17).

Example 1

In order to illustrate the operation of the linear programming algorithm let us consider a simple system described by the following inequalities

$$\begin{aligned} -1 &< -x_1 + x_2 < 1 \\ 1 &< x_1 + x_2 < 3 \\ 0.5 &< x_2 < 3 \end{aligned}$$

These are depicted in Figure 2. Clearly, for the linear system the linearization error is zero (thus there is no need for \mathbf{dx} and \mathbf{dz} notation). Consequently it is expected that the linear programming method will calculate the bounding box on the state uncertainty set, $\mathbf{X}^*([\mathbf{z}])$, that is identical to the one produced by the Monte Carlo simulations (given a sufficiently large number of Monte Carlo iterations).

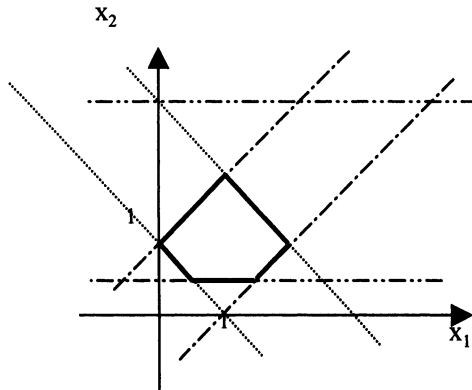


Figure 2. State uncertainty set defined by three double inequalities.

We select the first two inequalities as the minimum measurement set, thus defining $(z^n)'$, $(z^n)''$, $(z^{m-n})'$, $(z^{m-n})''$, J^n and J^{m-n} .

$$(z^n)' = [-1 \ 1]^T, (z^n)'' = [1 \ 3]^T, (z^{m-n})' = [0.5]^T, (z^{m-n})'' = [3]^T,$$

$$J^n = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix} \text{ and } J^{m-n} = [0 \ 1]$$

So, the corresponding matrices $(J^n)^{-1}$ and $J^{m-n}(J^n)^{-1}$ are

$$(J^n)^{-1} = \begin{bmatrix} -0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \text{ and } J^{m-n}(J^n)^{-1} = [0.5 \ 0.5]$$

The cost function, $a^i \cdot z^n$, for $i=1$ is $a^1 \cdot z^n = -0.5z_1^n + 0.5z_2^n$, so minimizing it with respect of z^n , subject to constraints (20), gives

$$(z_1^n)^- = 1 \text{ and } (z_2^n)^- = 1$$

and maximizing the cost function gives

$$(z_1^n)^+ = -1 \text{ and } (z_2^n)^+ = 3$$

For $i=2$ we minimize $a^2 \cdot z^n = 0.5z_1^n + 0.5z_2^n$, to obtain

$$(z_1^n)^- = -1+s \text{ and } (z_2^n)^- = 2-s$$

(where the parameter s , $0 \leq s \leq 1$, signifies the degenerate LP case)

The maximization of $a^2 \cdot z^n$ gives

$$(z_1^n)^+ = 1 \text{ and } (z_2^n)^+ = 3$$

Evaluating now the equations $x_i^- = a^i \cdot (z^n)^-$, and $x_i^+ = a^i \cdot (z^n)^+$, for $i=1$ and $i=2$ we have

$$\begin{aligned} x_1^- &= a^1 \cdot z^n = [-0.5 \ 0.5][1 \ 1]^T = 0, \\ x_1^+ &= a^1 \cdot z^n = [-0.5 \ 0.5][-1 \ 3]^T = 2, \\ x_2^- &= a^2 \cdot z^n = [0.5 \ 0.5][-1+s \ 2-s]^T = 0.5 \\ x_2^+ &= a^2 \cdot z^n = [0.5 \ 0.5][1 \ 3]^T = 2, \end{aligned}$$

which is the exact solution: $0 \leq x_1 \leq 2$ and $0.5 \leq x_2 \leq 2$.

Ellipsoid Method

An alternative to linear programming based confidence limit analysis is a method based on the iterative shrinking ellipsoids. The method has been first reported in (Schweppe, 1973) and has been considered by other researchers eg (Fogel, 1982; Norton, 1986; Bargiela, 1994). The technique is referred here as *ellipsoid method*.

Mathematically, an ellipsoid, E^t is a region of space defined as follows:

$$E^t := \{x \in R^n: (x-x^t)^T P_t^{-1} (x-x^t) \leq 1.0\} \quad (22)$$

for some $x^t \in R^n$ and some symmetric and positive-definite matrix P_t , of dimension n by n . E^t is therefore, a region of R^n centered on x^t . The aim of the ellipsoid method is to start with a large ellipsoid (usually a n -dimensional sphere) that contains the whole state uncertainty set, and then to generate a sequence of ellipsoids, decreasing in size, leading to one that fits the state uncertainty set as tightly as possible. Using an ellipsoid as an approximation to the state uncertainty set provides a simple and concise description of what can be a very complicated set. The algorithm itself, has the advantages of being sequential, mathematically and conceptually simple and can be very fast computationally.

We consider here the application of the Schweppe's ellipsoid algorithm to confidence limit analysis. The first point to note is that it is a linear method and so the state uncertainty set to be approximated is the set $X^1([z])$, for a measurement set M and measurement vector z^0 . An ellipsoid that is certain to contain $X^1([z])$ is used as the starting ellipsoid E^t . This ellipsoid may be the n -dimensional sphere centered at the state estimate \hat{x} (this is the state estimate generated by z^0) with a suitably large diameter, α . In this case $P_0 = \alpha.I$, where I is the n by n identity matrix. The 'observations' in confidence limit analysis are the linearized measurement constraints provided by (10), which can be re-written as

$$z^- - g(\hat{x}) + J. \hat{x} \leq J.d x \leq z^+ - g(\hat{x}) + J. \hat{x} \quad (23)$$

for all x in $X^1([z])$. In this equation, $z^- - g(\hat{x}) + J. \hat{x}$ and $z^+ - g(\hat{x}) + J. \hat{x}$ are constant vectors and so can be pre-calculated. Expression (23) represents m constraints, bounding $J.d x$ above and below. Each of these is taken in turn and used to modify the current ellipsoid.

Suppose that the t^{th} constraint is being used to update the $t-1^{st}$ ellipsoid, $t \in \{2, \dots, m\}$, and that E^{t-1} contains $X^1([z])$. The region F^t bounded by this constraint also contains the state uncertainty set $X^1([z])$. So $X^1([z])$ is contained in the intersection of these two regions as shown in Figure 3. A new ellipsoid, E^t , can be produced which

contains the intersection of \mathbf{E}^{t-1} and the region \mathbf{F}^t bounded by the constraint's hyperplanes. \mathbf{E}^t is the ellipsoid $\{\mathbf{x} \in \mathbf{R}^n: (\mathbf{x} - \mathbf{x}^t)^T P_t^{-1} (\mathbf{x} - \mathbf{x}^t) \leq 1.0\}$, where

$$\mathbf{x}^t = \mathbf{x}^{t-1} + (\rho_t \mathbf{v}_t / (e_t^z)^2) P_{t-1}' \mathbf{a}^t \quad (24)$$

$$P_t = (1 + \rho_t - (\rho_t \mathbf{v}_t / ((e_t^z)^2 + \rho_t g_t))) P_{t-1}' \quad (25)$$

$$P_{t-1}' = (I + (\rho_t / ((e_t^z)^2 + \rho_t g_t)) P_{t-1} \mathbf{a}^t (\mathbf{a}^t)^T) P_{t-1} \quad (26)$$

$$\mathbf{g}_t = (\mathbf{a}^t)^T P_{t-1} \mathbf{a}^t \quad (27)$$

$$\mathbf{v}_t = 0.5(\mathbf{z}_t^+ + \mathbf{z}_t^-) - (\mathbf{g}(\mathbf{x}^{t-1}))_t + (J \mathbf{d}\mathbf{x})_t + (\mathbf{a}^t)^T \mathbf{x}^{t-1} \quad (28)$$

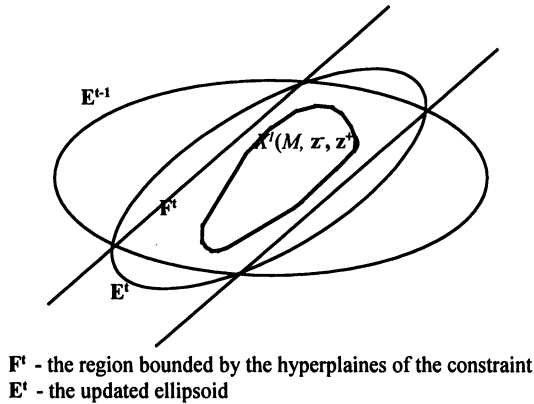


Figure 3 Ellipsoid update (in 2-dimensions).

In these equations, P_{t-1} and \mathbf{x}^{t-1} are the positive definite matrix and center vector, respectively, for the previous ellipsoid, \mathbf{E}^{t-1} , and ρ_t can be any non-negative real value. The value, e_t^z used in these equations, is the t^{th} element of the measurement error vector, $\mathbf{e}^z = \mathbf{z}^+ - \mathbf{z} = \mathbf{z} - \mathbf{z}^- = 0.5(\mathbf{z}^+ - \mathbf{z}^-)$. It should be noted that, despite the fact that (22) refers to P_t^{-1} matrix and (24) to (28) refer to P_t , no matrix inversion is involved in the algorithm. In fact, the matrix P_t^{-1} need never be known as all updating is performed using matrices P_{t-1} and P_t .

The choice of the parameter ρ_t gives rise to various ellipsoid methods. In (Fogel, 1982) there are two suggestions. The first involves the solution of a quadratic equation in ρ_t and produces the ellipsoid of minimum volume. The second requires the solution of a cubic equation and minimizes the sum of squares of the semi-axis in \mathbf{E}^t . Alternative choices are discussed in depth in (Milanese, 1996).

On termination of the algorithm, the confidence limits for each variable are easily calculated from the final positive-definite matrix, P_n , and the final center vector \mathbf{x}^t . These are

$$x_i^{1+} = x_i' + \sqrt{P_i(i,i)} \quad (29)$$

$$x_i^{1-} = x_i' - \sqrt{P_i(i,i)} \quad (30)$$

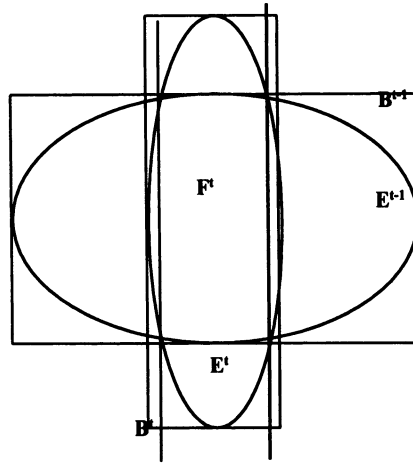
Ellipsoidal confidence limit algorithm

1. Set $t=0$ and $P_0=\alpha I$.
2. Set $t=t+1$.
3. Calculate g_t and v_t from equations (27) and (28).
4. Find ρ_t that minimizes the volume of the new ellipsoid by solving the following quadratic equation in ρ_t :

$$(p-1)g_t^2 \cdot \rho_t^2 + ((2p-1) \cdot (e_t^z)^2 - g_t \cdot v_t)g_t \cdot \rho_t + (e_t^z)^2(p \cdot ((e_t^z)^2 - v_t^2) - g_t) = 0$$
5. Calculate P'_{t-1} from (26).
6. Update the state variable x^t , as per equation (24).
7. Update P_t , equation (25).
8. If not all constraints have been processed yet then repeat from step 2.
9. If the volume of the ellipsoid has been reduced by less than a pre-specified ratio than stop, otherwise reset the constraints counter, $t=0$, and repeat from step 2.

In some situations, tight bounds can be found by processing each measurement constraint only once, in which case only m steps are required. However, published research suggests that further reduction in the bounds is often possible by re-processing some or all of the constraints (Belforte, 1985). Also, the variation of the order in which the constraints are processed has the effect on the rate of convergence of the algorithm.

Although the computational complexity of the ellipsoid algorithm compares favorably with the linear programming method, its effectiveness in bounding the state uncertainty set can be poor. The reason for that is that when the Jacobian, J , of the uncertain system equation, $g(\cdot)$ is sparse, each measurement constraint only bounds a few of the variables. In the ellipsoid algorithm, constraints are considered individually and so can only improve confidence limits on the few variables that they bound explicitly. Using a 2-dimensional example in which the observation hyperplanes each constrain only one of the variables (Figure 4) we can see that the new ellipsoid, produces tighter confidence limits in the horizontal direction but looser ones in the vertical direction. When this idea is extended to many dimensions, only a few of which are bound by each constraint, it is easy to see that at each iteration the majority of the variables will have their confidence limits increased and only a minority will have them reduced.



The new bounds, marked by B^t , are tighter than B^{t-1} in the horizontal direction but are not as tight in the vertical direction

Figure 4. Ellipsoid update does not always lead to improvement in all variables.

Example 2

The system of inequalities introduced in Example 1 is now processed using the ellipsoid algorithm. As in the previous three examples we note that for the linear system the dx variable is identical to x so the equation (28) simplifies to

$$v_t = 0.5(z_t^+ + z_t^-) + (a^t)^T \cdot x^{t-1}.$$

The initial ellipsoid is assumed to be a hyper sphere of radius $\sqrt{10}$ centered at $x^0 = [0, 0]^T$. The measurement upper- and lower- bound vectors are $z^- = [-1 \ 1 \ 0.5]^T$ and $z^+ = [1 \ 3 \ 3]^T$ and the matrices J and P_0 are

$$J = \begin{bmatrix} -1 & 1 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \text{ and } P_0 = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

Processing the three constraints in the order $t=1,2,3$ we obtain, rounded to two places after decimal point

$$g_1=20.00 \quad v_1=0.00 \quad P_1 = \begin{bmatrix} 7.26 & 5.09 \\ 5.09 & 7.26 \end{bmatrix} \quad x^{1-} = \begin{bmatrix} -2.69 \\ -2.69 \end{bmatrix} \quad x^{1+} = \begin{bmatrix} 2.69 \\ 2.69 \end{bmatrix}$$

$$g_2=24.70 \quad v_2=2.00 \quad P_2=\begin{bmatrix} 2.18 & -0.21 \\ -0.21 & 2.18 \end{bmatrix} \quad \mathbf{x}^{2-}=\begin{bmatrix} -0.62 \\ -0.62 \end{bmatrix} \quad \mathbf{x}^{2+}=\begin{bmatrix} 2.33 \\ 2.33 \end{bmatrix}$$

$$g_3=2.18 \quad v_3=0.89 \quad P_3=\begin{bmatrix} 2.48 & -0.18 \\ -0.18 & 2.48 \end{bmatrix} \quad \mathbf{x}^{3-}=\begin{bmatrix} -0.74 \\ -0.30 \end{bmatrix} \quad \mathbf{x}^{3+}=\begin{bmatrix} 2.41 \\ 2.44 \end{bmatrix}$$

and reversing the order of processing of the constraints to $t=3,2,1$ gives

$$g_3=10.00 \quad v_3=1.75 \quad P_3=\begin{bmatrix} 12.33 & 0.00 \\ 0.00 & 3.12 \end{bmatrix} \quad \mathbf{x}^{3-}=\begin{bmatrix} -3.51 \\ -0.46 \end{bmatrix} \quad \mathbf{x}^{3+}=\begin{bmatrix} 3.51 \\ 3.07 \end{bmatrix}$$

$$g_2=15.44 \quad v_2=0.69 \quad P_2=\begin{bmatrix} 5.46 & -2.51 \\ -2.51 & 3.25 \end{bmatrix} \quad \mathbf{x}^{2-}=\begin{bmatrix} -1.89 \\ -0.38 \end{bmatrix} \quad \mathbf{x}^{2+}=\begin{bmatrix} 2.78 \\ 3.22 \end{bmatrix}$$

$$g_1=13.73 \quad v_1=-0.97 \quad P_1=\begin{bmatrix} 2.17 & 0.22 \\ 0.22 & 1.63 \end{bmatrix} \quad \mathbf{x}^{1-}=\begin{bmatrix} -0.57 \\ -0.18 \end{bmatrix} \quad \mathbf{x}^{1+}=\begin{bmatrix} 2.37 \\ 2.37 \end{bmatrix}$$

We note that the quality of bounding the state variables by the ellipsoid method is clearly inferior to that produced by the linear method. Furthermore a worrying characteristic of the ellipsoid technique is that it depends on the order of processing of constraints and there is no guidance on which order of processing of constraints should be chosen to ensure the most rapid convergence of the estimates. It is also worth pointing out that the criterion of non-increasing the volume of the ellipse, when progressing from one iteration to the next, does not necessarily imply the tightening of the bounds on all state variables, as can be seen from the transition from $t=2$ to $t=3$ in the example above. The consecutive ellipsoids generated in the example above are given in Figure 5.

Against this background a number of hybrid techniques combining the ellipsoid and linear programming methods have been proposed (Norton, 1986; Milanese, 1996). These primarily involve a fast pre-processing of constraints using ellipsoid method followed by the linear programming optimization with the reduced constraint set. In the subsequent sub-section we introduce an alternative hybrid technique that combines the point estimation of the average system state with linear programming estimation of the state uncertainty set.

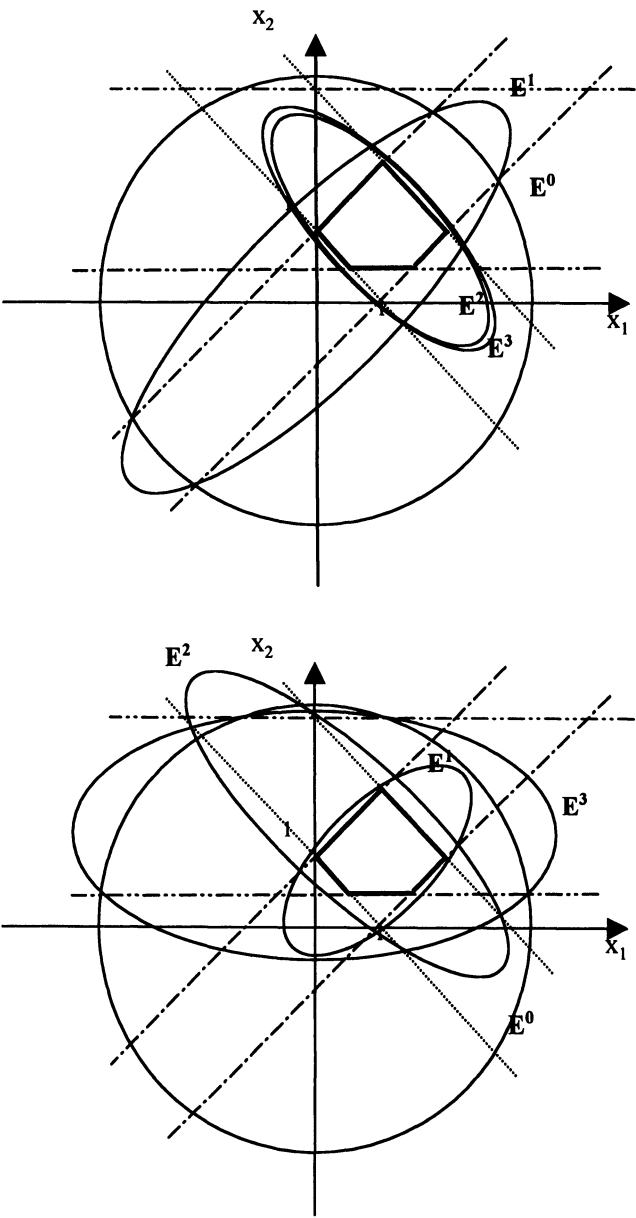


Figure 5. Confidence limits ellipsoids generated by processing the constraints in the order $t=1, 2, 3$ and $t=3, 2, 1$ respectively.

Sensitivity Matrix Method

The basis of the method introduced here is the observation that when the measurement set is minimal (ie it is observable and contains no observable subset), the linearized uncertainty bounds can be calculated without recourse to a linear programming procedure. In these circumstances, confidence limit analysis can be carried out much more rapidly than in the general case when the linear programming algorithm (even in its improved form (20), (21)) is used. This is possible because when M is minimal the Jacobian matrix J is square and invertible. So, any $\mathbf{dx} \in DX^1([z])$ is given by $J^{-1} \cdot \mathbf{dz}$ for some $\mathbf{dz} \in DZ^1([z])$. In general however, M is over-determined and so J is an m by n matrix of rank n . When J is of this form it has no inverse and \mathbf{dx} is calculated using pseudo-inverse as $\mathbf{dx} = (J^T J)^{-1} J^T \cdot \mathbf{dz}$. The matrix $(J^T J)^{-1} J^T$ is referred to as the *sensitivity matrix* as its $(i,j)^{th}$ element relates the sensitivity of the i^{th} element of the state vector to changes in the j^{th} element of the measurement vector.

A new approximate linearized state uncertainty set, $X^2([z])$, can be defined as follows:

$$X^2([z]) := \{\mathbf{x} \in \mathbf{R}^n: \mathbf{x} = \hat{\mathbf{x}} + \mathbf{dx}, \mathbf{dx} = (J^T J)^{-1} J^T \cdot \mathbf{dz}, \mathbf{dz} \in DZ([z])\} \quad (31)$$

Looking at the definition of the set $X^1([z])$, (9), it is easy to see that

$$X^1([z]) \subseteq X^2([z]) \quad (32)$$

and when M is a minimal observable measurement set

$$X^1([z]) = X^2([z]) \quad (33)$$

When M is over-determined there may be vectors $\mathbf{dz} \in DZ([z])$ that are inconsistent. For such a vector there can be no $\mathbf{dx} \in \mathbf{R}^n$ with $J \cdot \mathbf{dx} = \mathbf{dz}$. So, in particular $J(J^T J)^{-1} J^T \cdot \mathbf{dz}$ is not equal to \mathbf{dz} . It may even be that $J(J^T J)^{-1} J^T \cdot \mathbf{dz}$ is not contained in $DZ([z])$. It is these vectors that account for the difference between $X^1([z])$ and $X^2([z])$. That is, $\mathbf{dx} + \hat{\mathbf{x}} \in X^2([z]) - X^1([z])$ if and only if the state vector $\mathbf{dx} = (J^T J)^{-1} J^T \cdot \mathbf{dz}$ for some $\mathbf{dz} \in DZ([z])$ with $J(J^T J)^{-1} J^T \cdot \mathbf{dz}$ not a member of $DZ([z])$. Although $X^2([z])$ is not identical to $X^1([z])$ it can be used to form bounds for the linearized state uncertainty set. Although these bounds are less tight than the ones obtained with the linear programming method, at least they do not rule out any feasible state vector from the uncertainty box.

Bounding vectors for $X^2([z])$, denoted \mathbf{x}^{2-} and \mathbf{x}^{2+} , represent a can be defined analogously to \mathbf{x}^{1-} and \mathbf{x}^{1+} for the true linearized state uncertainty set $X^1([z])$. The following algorithm provides a way of calculating these vectors.

Sensitivity matrix confidence limit algorithm

1. Set $i = 0$.
2. Factorize the matrix $J^T J$ (This can be done using an augmented matrix formulation so as to preserve the condition number of the matrix J).
3. Set $i = i + 1$.
4. Calculate \mathbf{b}^i , the i^{th} row of the sensitivity matrix $(J^T J)^{-1} J^T$ (This can be done efficiently taking into account the sparsity of J and using the augmented matrix based factorization of step 2).
5. Put $\mathbf{x}_i^{2+} = \mathbf{b}^i \cdot \mathbf{dz}^+ + \hat{\mathbf{x}}_i$, where

$$\mathbf{dz}_j^+ = \begin{cases} \mathbf{dz}_j^+ & \text{if } b_j^i > 0.0 \\ \mathbf{dz}_j^- & \text{otherwise} \end{cases} \quad (34)$$

Put $\mathbf{x}_i^{2-} = \mathbf{b}^i \cdot \mathbf{dz}^- + \hat{\mathbf{x}}_i$, where

$$\mathbf{dz}_j^- = \begin{cases} \mathbf{dz}_j^- & \text{if } b_j^i > 0.0 \\ \mathbf{dz}_j^+ & \text{otherwise} \end{cases} \quad (35)$$

7. If $i < n$, go back to step 3. Otherwise stop.

The computational complexity of the above algorithm is of order $O(n^2 + nm)$. For a large values of n and m this offers significant computational savings compared to linear programming optimization. For example if $n=200$ and $m=300$, the computational complexity of the sensitivity matrix method is of order $O(10^5)$, while the computational complexity of the linear programming method is of order $O(4 \cdot 10^6)$.

Example 3

We apply here the sensitivity matrix confidence limit algorithm to the system of inequalities considered in Example 1.

The pseudoinverse matrix $(J^T J)^{-1} J^T$ is

$$(J^T J)^{-1} J^T = \begin{bmatrix} -0.5 & 0.5 & 0 \\ 0.333 & 0.333 & 0.333 \end{bmatrix}$$

so, $\mathbf{b}^1 = [-0.5 \ 0.5 \ 0]$ and $\mathbf{b}^2 = [0.333 \ 0.333 \ 0.333]$.

Processing \mathbf{z} , according to (34) and (35), using the first row, \mathbf{b}^1 , we obtain $\mathbf{z}^- = [1.0 \ 1.0 \ 0.5]^T$ and $\mathbf{z}^+ = [-1.0 \ 3.0 \ 3.0]^T$ which produces

$$\mathbf{x}_1^l = \mathbf{b}^1 \mathbf{z}^- = [-0.5 \ 0.5 \ 0] [1.0 \ 1.0 \ 0.5]^T = 0$$

$$x_1^u = \mathbf{b}^1 \mathbf{z}^+ = [-0.5 \ 0.5 \ 0] [-1.0 \ 3.0 \ 3.0]^T = 2$$

and using the second row, \mathbf{b}^2 , we have $\mathbf{z}^- = [-1.0 \ 1.0 \ 0.5]^T$ and $\mathbf{z}^+ = [1.0 \ 3.0 \ 3.0]^T$ which gives

$$x_2^l = \mathbf{b}^2 \mathbf{z}^- = [0.333 \ 0.333 \ 0.333] [-1.0 \ 1.0 \ 0.5]^T = 0.167$$

$$x_2^u = \mathbf{b}^2 \mathbf{z}^+ = [0.333 \ 0.333 \ 0.333] [1.0 \ 3.0 \ 3.0]^T = 2.333$$

The bounding box on the state uncertainty set, $0 \leq x_1 \leq 2$ and $0.167 \leq x_2 \leq 2.33$, as calculated here, is larger than the one obtained with the Monte Carlo and the linear programming algorithm. The widening of bounds along the x_2 direction is caused by the inherent feature of the pseudoinverse, that of attempting to balance the sum of distances from x_i^- and x_i^+ to all upper- and lower- bound constraints respectively. By contrast, the linear programming and the Monte Carlo algorithms are concerned only with the 'active' constraints for any given value of the state vector, thus ignoring the redundant constraints.

Interval arithmetic formalism for the sensitivity matrix method

The computation of bounds on individual state variables, implemented in Step 5 of the above algorithm (equations (34) and (35)), can be expressed using the formalism of interval arithmetic introduced in Chapter 2. Using this formalism we can express the computation of hyperbox enclosure of $X^2(\mathbf{z})$ as

$$[x_i] = [\mathbf{b}^i] \cdot [\mathbf{dz}] + [\hat{x}_i] \quad , \quad i=1, \dots, n \quad (36)$$

where $[\mathbf{b}^i]$ is a vector point-interval formed by the i^{th} row of the sensitivity matrix $(\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T$ and $[\hat{x}_i]$ is a point-interval $[\hat{x}_i, \hat{x}_i]$.

Example 4

Continuing with the system of inequalities considered in Example 1 we can formulate the interval equations for the sensitivity matrix method. The point-interval vectors corresponding to the two rows of the pseudoinverse matrix are

$$\begin{aligned} [\mathbf{b}^1] &= [[-0.5, -0.5] \ [0.5, 0.5] \ [0, 0]] \\ [\mathbf{b}^2] &= [[0.333, 0.333] \ [0.333, 0.333] \ [0.333, 0.333]] \end{aligned}$$

and the interval vector \mathbf{z} is

$$[\mathbf{z}] = [[-1, 1] \ [1, 3] \ [0.5, 3]]$$

Using $[\mathbf{b}^1]$ and $[\mathbf{z}]$ we can calculate the interval $[x_1]$ as follows

$$[x_1] = [b^1] \cdot [z] = [-0.5, -0.5][-1, 1] + [0.5, 0.5][1, 3] + [0, 0][0.5, 3] = [-0.5, 0.5] + [0.5, 1.5] + [0, 0] = [0, 2]$$

and

$$[x_2] = [b^2] \cdot [z] = [0.333, 0.333][-1, 1] + [0.333, 0.333][1, 3] + [0.333, 0.333][0.5, 3] = [-0.333, 0.333] + [0.333, 1.0] + [0.167, 1.0] = [0.167, 2.333]$$

As expected, the result is identical to that calculated by the sensitivity matrix method (Example 3). However, it must be borne in mind that the mathematical elegance of the interval arithmetic is paid for with the increased computational burden. While the interval arithmetic calculations, listed above, required 48 multiplications, 48 logical operations and 8 additions, those in Example 3 were accomplished with 12 multiplications, 12 logical operations and 8 additions.

18. 3 REAL-LIFE APPLICATION

The uncertainty models from the previous section are illustrated here in the context of state estimation of water distribution networks. A network represented diagrammatically in Figure 6 consists of 65 nodes, 92 pipes and 5 inflow points. The inflow points are the reservoirs at nodes 60 and 160, a pumping station at node 68 and two water supplies from a high-pressure zone through pressure-reducing valves at nodes 3 and 26. Pipe data: - length [m]; - diameter [m]; and C-values (conductivity); are listed in Table 1. This data, together with the reference pressure measurement in node 160 and five inflow measurements in nodes 3, 26, 60, 68 and 160 allows calculation of the system state (a 70-dimensional vector of 65 nodal pressures and 5 inflows in fixed-pressure nodes).

Two measurement sets are considered, both for the same operating state. The first measurement set, *M1*, is a minimal measurement set, consisting of nodal consumption values in all but one of the nodes, an inflow measurement for each of the inflow nodes and one reference pressure measurement at node 160. The second measurement set, *M2*, consists of all measurements contained in the set *M1* together with 4 additional pressure and 4 flow measurements.

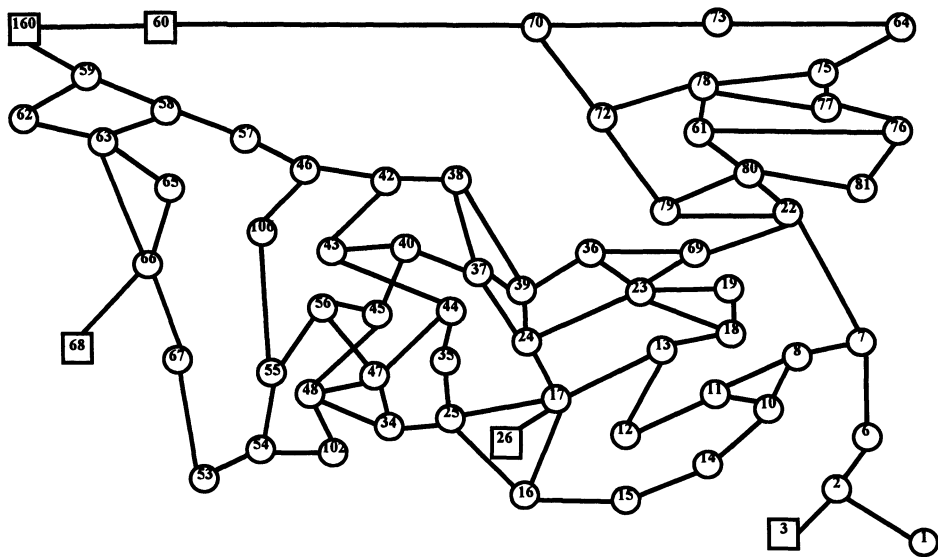


Figure 6. Water distribution network used for testing the confidence limits calculations.

Table 1. Pipe data for the test network.

<i>Pipe</i>	<i>Len.</i>	<i>Dia.</i>	<i>C</i>	<i>Pipe</i>	<i>Len.</i>	<i>Dia.</i>	<i>C</i>	<i>Pipe</i>	<i>Len.</i>	<i>Dia.</i>	<i>C</i>
1-2	800	0.200	140	23-24	300	0.300	145	43-40	380	0.250	80
2-3	400	0.300	165	24-17	650	0.200	158	40-45	580	0.168	120
2-6	400	0.300	165	17-25	330	0.175	127	40-37	320	0.168	120
6-7	970	0.300	165	25-16	630	0.225	104	37-24	390	0.200	145
7-8	300	0.225	135	26-17	360	0.300	80	37-39	280	0.168	120
8-10	350	0.225	171	25-34	780	0.175	80	39-24	300	0.150	90
8-11	350	0.125	60	25-35	320	0.225	119	37-38	270	0.200	145
11-10	360	0.150	105	35-44	710	0.225	90	38-39	550	0.300	229
11-12	180	0.125	60	44-47	520	0.250	70	39-36	210	0.300	127
12-13	200	0.175	115	47-34	610	0.225	145	36-23	180	0.300	112
10-14	710	0.225	110	47-48	540	0.250	70	36-69	147	0.300	145
14-15	225	0.225	110	48-34	900	0.175	80	22-79	160	0.225	80
15-16	310	0.225	90	48-102	310	0.250	70	79-80	340	0.150	90
16-17	590	0.094	80	102-54	660	0.225	140	80-22	390	0.200	145
17-13	740	0.175	115	54-53	480	0.225	158	80-81	1220	0.150	139
13-18	250	0.225	158	54-55	380	0.225	159	81-76	600	0.150	145
18-19	330	0.300	145	55-56	190	0.225	145	76-77	670	0.150	116
22-7	1510	0.225	96	56-45	610	0.125	55	77-78	150	0.150	116
22-69	120	0.300	145	45-48	1060	0.175	80	78-61	460	0.094	170
69-23	420	0.150	90	44-43	230	0.250	80	61-80	530	0.150	145

72-78	1100	0.142	105	59-58	630	0.300	50	66-63	800	0.225	110
72-79	600	0.225	60	58-57	730	0.300	118	66-65	210	0.125	60
72-70	1770	0.225	47	57-46	260	0.300	85	65-63	590	0.125	60
70-73	3090	0.356	46	46-42	250	0.300	85	63-58	2050	0.150	40
73-64	410	0.225	80	42-38	430	0.300	145	63-62	770	0.225	110
64-75	420	0.225	80	42-43	720	0.250	80	62-59	350	0.225	110
75-78	350	0.094	170	46-106	720	0.200	145	23-19	430	0.300	145
75-77	400	0.150	145	106-55	700	0.142	137	23-18	760	0.117	60
61-76	1470	0.150	81	56-47	550	0.225	145	66-68	440	0.300	170
70-60	2200	0.356	100	53-67	220	0.225	160	60-160	270	0.381	32
160-59	370	0.381	50	67-66	270	0.225	160				

As it is pointed out in Section 1.1, the true measurement vector, \mathbf{z}^t , rarely coincides with the observed one, \mathbf{z}^o . The discrepancy is caused by meter noise affecting real measurements and the inaccuracy of estimates, which are used as pseudomeasurements. In order to reflect this reality the observed measurement values, \mathbf{z}^o , are generated in the following way. Firstly, a true operating state, \mathbf{x}^t , listed in column 2 of Table 3, is assumed and a true measurement vector, \mathbf{z}^t , is calculated as $\mathbf{g}(\mathbf{x}^t)$. The true measurement values, \mathbf{z}^t , are listed in column 2 of Table 2. The observed measurement values, \mathbf{z}^o , listed in column 3, are selected randomly from within the range $[\mathbf{z}^-, \mathbf{z}^+]$, in accordance with (2). The bounds $[\mathbf{z}^-, \mathbf{z}^+]$ were defined in terms of relative variability of \mathbf{z}^t as follows: 50% for $\mathbf{z}^t < 0.5$ l/s; 40% for $0.5 < \mathbf{z}^t < 1.0$ l/s; 30% for $1.0 < \mathbf{z}^t < 5.0$ l/s; 20% for $5.0 < \mathbf{z}^t < 10.0$ l/s and 10% for $\mathbf{z}^t > 10.0$ l/s. This corresponds to real-life situation where measurement values are not exact but are contained within the range specified by the accuracy of meters. The state vector, $\hat{\mathbf{x}}$, calculated from the observed measurement vector, \mathbf{z}^o , is shown in column 3 of Table 3. The difference between this state estimate and the true state should be noted. It is caused solely by the addition of the simulated measurement errors and shows how noise corrupted measurement data affects system state.

Table 2. Measurement data.

<i>Node</i>	<i>True</i>	<i>Observed</i>	<i>Node</i>	<i>True</i>	<i>Observed</i>	<i>Node</i>	<i>True</i>	<i>Observed</i>
<i>Reference pressure [m]</i>			7	1.64	0.98	24	1.73	2.01
160	144.77	144.75	8	1.16	0.78	25	2.71	2.21
<i>Inflows [l/s]</i>			10	9.64	8.56	61	0.42	0.22
26	65.00	65.54	11	0.34	0.29	34	7.40	5.96
3	31.00	31.43	12	0.35	0.53	35	2.58	2.48
60	34.00	33.46	13	0.50	0.79	36	1.92	2.44
160	45.00	45.85	14	6.54	5.38	37	2.98	2.20
68	31.00	30.93	15	3.18	4.40	38	2.36	1.64
<i>Consumptions [l/s]</i>			16	2.01	1.46	39	0.65	1.00
1	4.85	5.00	17	8.51	10.27	40	6.77	7.48
2	6.77	5.65	18	8.71	9.69	102	2.13	2.14
6	2.09	1.94	19	0.00	0.00	42	8.03	7.68
			22	2.35	2.36	43	3.51	4.56
			23	0.47	0.49	44	1.89	1.97

45	1.10	1.62	66	2.15	2.76	3	7.95	8.91
46	2.73	3.19	67	1.87	2.45	60	0.58	0.79
47	10.80	12.82	69	4.52	6.21	68	2.46	2.55
48	2.95	2.47	70	2.18	1.74	<i>M2 – pressure measurements</i>		
53	0.67	0.84	72	11.51	11.70	7	140.08	140.08
54	4.54	4.79	73	2.77	2.47	44	139.85	139.85
55	10.83	9.47	75	1.32	1.39	66	141.42	141.42
56	0.78	0.85	76	5.37	4.83	80	140.10	140.10
57	0.16	0.12	77	1.16	0.93	<i>M2 - flow measurements</i>		
58	5.68	4.38	78	1.35	1.27	22-69	-7.13	-7.13
59	2.88	2.92	79	1.91	2.52	42-38	-0.16	-0.16
62	2.94	3.66	80	2.64	2.56	7-22	1.94	1.94
63	10.46	10.54	81	2.79	2.21	56-45	0.30	0.30
64	3.75	2.99	106	1.74	2.55			
65	3.84	3.99	26	0.26	0.31			

Table 3. True and estimated state vector.

<i>StateNode</i>	<i>TrueState</i>	<i>Estimate</i>	<i>StateNode</i>	<i>TrueState</i>	<i>Estimate</i>	<i>StateNode</i>	<i>TrueState</i>	<i>Estimate</i>
(1)-1	140.11	140.04	(25)-38	140.33	140.15	(49)-69	140.31	140.14
(2)-2	140.23	140.17	(26)-39	140.33	140.15	(50)-70	143.88	143.90
(3)-6	140.20	140.14	(27)-40	140.06	139.84	(51)-72	140.25	140.10
(4)-7	140.15	140.08	(28)-102	140.07	139.86	(52)-73	141.78	141.87
(5)-8	140.02	139.96	(29)-42	140.33	140.15	(53)-75	140.73	140.76
(6)-10	139.94	139.89	(30)-43	140.07	139.85	(54)-76	139.97	139.95
(7)-11	140.02	139.95	(31)-44	140.06	139.85	(55)-77	140.36	140.34
(8)-12	140.38	140.21	(32)-45	140.02	139.80	(56)-78	140.35	140.32
(9)-13	140.41	140.23	(33)-46	140.45	140.27	(57)-79	140.27	140.11
(10)-14	139.91	139.84	(34)-47	139.96	139.75	(58)-80	140.24	140.10
(11)-15	139.93	139.85	(35)-48	139.99	139.78	(59)-81	139.97	139.95
(12)-16	140.05	139.95	(36)-53	140.92	140.65	(60)-106	140.34	140.13
(13)-17	141.81	141.58	(37)-54	140.21	139.99	(61)-26	144.37	144.18
(14)-18	140.36	140.18	(38)-55	140.05	139.85	(62)-3	140.34	140.28
(15)-19	140.36	140.18	(39)-56	140.03	139.83	(63)-60	144.82	144.81
(16)-22	140.30	140.13	(40)-57	140.71	140.56	(64)-160	144.77	144.75
(17)-23	140.36	140.18	(41)-58	141.11	141.00	(65)-68	141.88	141.59
(18)-24	140.40	140.22	(42)-59	143.48	143.39	<i>Inflows [l/s]</i>		
(19)-25	140.24	140.09	(43)-62	142.84	142.66	(66)-26	65.00	65.54
(20)-61	140.23	140.10	(44)-63	141.79	141.53	(67)-3	31.00	31.43
(21)-34	139.94	139.74	(45)-64	141.11	141.20	(68)-60	34.00	33.46
(22)-35	140.17	140.00	(46)-65	141.32	141.01	(69)-160	45.00	45.85
(23)-36	140.33	140.15	(47)-66	141.71	141.42	(70)-68	31.00	30.95
(24)-37	140.32	140.14	(48)-67	141.25	140.97			

The first set of results concerns the state uncertainty sets, $X(M1, \bar{z}, z^+)$ and $X^1(M1, \bar{z}, z^+)$, for the minimal measurement set, $M1$, as calculated by the Monte Carlo confidence limit algorithm and the linear programming confidence limit algorithm. Results for the sensitivity matrix algorithm, $X^2(M1, \bar{z}, z^+)$, are not included because for a minimal measurement set these are identical to those using linear programming algorithm.

Rather than trying to visualize the state vectors themselves we focus our attention on their variability, $(x_i^+ - x_i^-)/2$, around the average value, $(x_i^+ + x_i^-)/2$, for each variable $i \in \{1, \dots, n\}$. Figure 7 depicts the following state uncertainty variability sets $\Delta X(M1, \bar{z}, z^+)$ and $\Delta X^1(M1, \bar{z}, z^+)$:

$$\Delta X(M1, \bar{z}, z^+) := \{ \delta x \in \mathbb{R}^{+n} : \delta x = |dx|, \hat{x} + dx \in X(M1, \bar{z}, z^+) \} \quad (41)$$

$$\Delta X^1(M1, \bar{z}, z^+) := \{ \delta x \in \mathbb{R}^{+n} : \delta x = |dx|, \hat{x} + dx \in X^1(M1, \bar{z}, z^+) \} \quad (42)$$

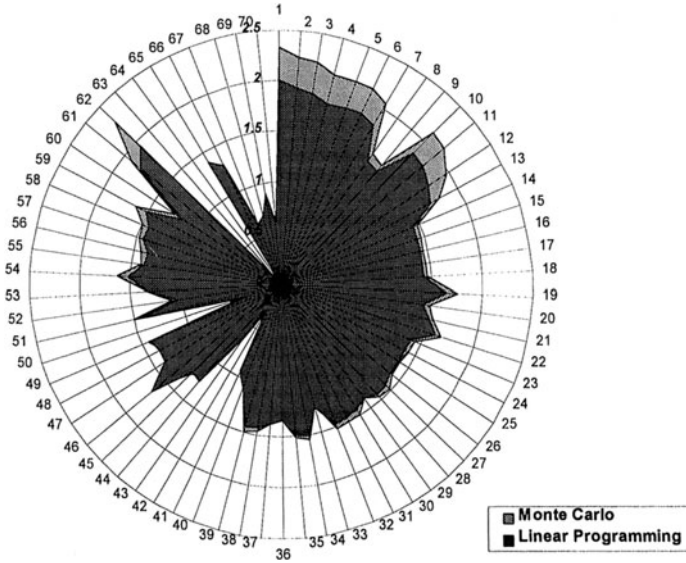


Figure 7. State uncertainty variability sets for Monte Carlo and Linear Programming methods.

The Monte Carlo and linear programming results demonstrate the scale of the potential error in state estimates for a system with no measurement redundancy.

Pressure errors are in excess of 2.0 [m] in the region of the network that is most distant from the reference pressure node. In fact, the majority of pressure errors are over 1.0 [m] with only those nodes close to node 160 having relatively tight uncertainty bounds. This indeed confirms the intuitive understanding of the relationship between the uncertainty bounds and the accuracy and location of measurements in the system.

The linear programming results correlate well with the Monte Carlo results; with the linear programming error bounds being no more than 5% off the bounds calculated by the Monte Carlo method for all but the most distant nodes from the reference pressure e.i. 1, 2, 3, 6, 7, 8, 10, 11, 14, 15, 16, 17 and 26, for which the discrepancy is less than 15% (the corresponding state variables are 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 61 and 62). These observations lead to the conclusion that no significant accuracy is lost in linearizing the uncertainty model and justify the use of linearized confidence limit algorithms.

The results obtained with the Ellipsoid method are difficult to assess objectively as they depend on the order of processing of the measurements. Figure 8 illustrates two sets of results obtained by varying the order of processing of the measurements. In either case it is clear that the Ellipsoid method produces results that are inferior to those obtained with the linear programming method.

The second set of results concerns the state uncertainty sets $X^1(M2, \bar{z}, z^+)$ and $X^2(M2, \bar{z}, z^+)$ calculated by the linear programming and the sensitivity matrix methods using the augmented measurement set, $M2$. Because of the huge computational requirements of the Monte Carlo method, when the increased number of measurements raises the chances of generating infeasible state vectors, MC method was deemed impractical and was not attempted. As with the previous measurement set, the results are analyzed in terms of the state uncertainty variability sets $\Delta X^1(M2, \bar{z}, z^+)$ and $\Delta X^2(M2, \bar{z}, z^+)$. These are presented in Figure 9.

$$\Delta X^1(M2, \bar{z}, z^+) := \{ \delta x \in \mathbf{R}^{+n} : \delta x = |dx|, \hat{x} + dx \in X^1(M2, \bar{z}, z^+) \} \quad (43)$$

$$\Delta X^2(M2, \bar{z}, z^+) := \{ \delta x \in \mathbf{R}^{+n} : \delta x = |dx|, \hat{x} + dx \in X^2(M2, \bar{z}, z^+) \} \quad (44)$$

Since for both methods the state estimate for a given measurement vector, z^0 , is \hat{x} , and since $\Delta X^1(M2, \bar{z}, z^+) \subseteq \Delta X^2(M2, \bar{z}, z^+)$, the results demonstrate the important result presented in (32) that $X^1(M2, \bar{z}, z^+) \subseteq X^2(M2, \bar{z}, z^+)$.

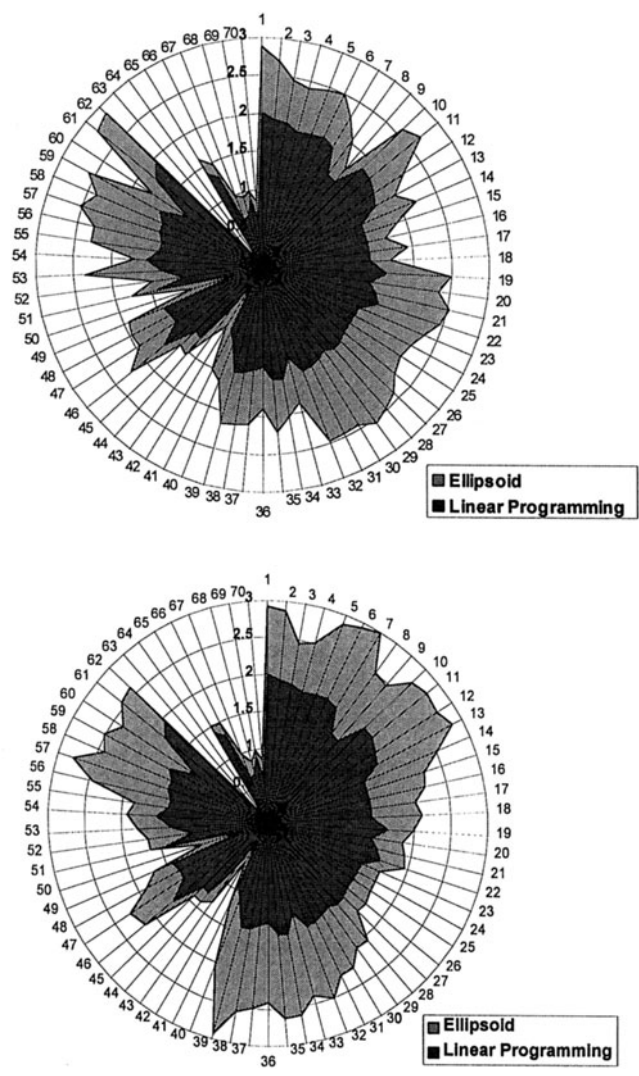


Figure 8. State uncertainty variability sets for two different runs of the Ellipsoid method compared to the result of the Linear Programming method.

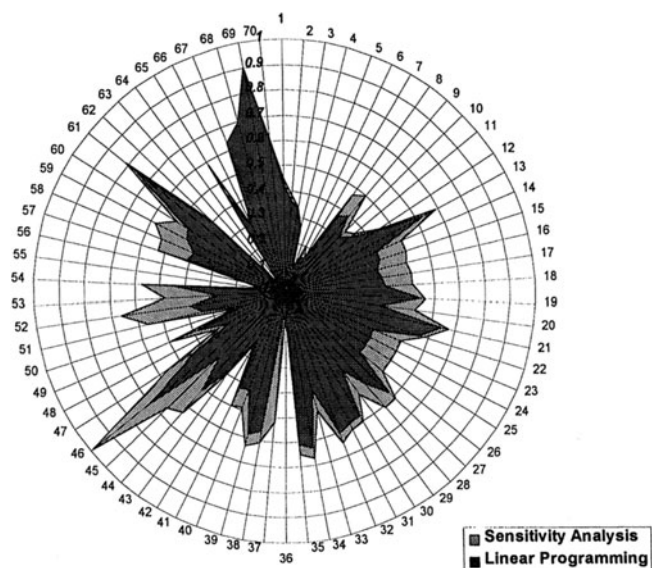


Figure 9. State uncertainty variability sets for Sensitivity Analysis and Linear Programming methods.

Considering the individual variables of the state vector it is clear that the linear programming and the sensitivity matrix results are closely correlated and that the overestimation of the state uncertainty variability set with the sensitivity matrix method is confined to approx. 30% of the range of the variables. Again it is interesting to see that the uncertainty bounds have been tightened considerably in the vicinity of the additional meters.

18. 4 CONCLUSIONS

State estimation of nonlinear systems is a challenging task that represents a large class of real-life problems. This is particularly so if a more realistic, interval representation of measurement uncertainty is adopted. Clearly, while the interval-estimation represents a positive development in terms of building abstract knowledge, its practical value depends on two crucial factors. The ability to produce the bounds on individual variables as tight as possible (thus maximizing the information content of interval-estimates) and the ability to calculate efficiently the results so that the method is compatible with real-time requirements of physical systems.

We have considered here one non-linear interval-estimation technique (Monte Carlo) and 3 linear techniques (linear programming, ellipsoidal bounding and sensitivity matrix) which have been applied to the linearized system model. The analysis shows that the linearisation has only a second-order effect on the shape of the state uncertainty set and, as such, it is an acceptable simplification leading to much more efficient estimation techniques.

The *Monte Carlo* estimation of the state uncertainty set is most accurate as it does not need to make any simplifying assumption about the nonlinear system model but it is computationally inefficient and consequently cannot be applied to any but the simplest systems. Of the three linear interval-estimators the *Linear Programming* technique produces tightest bounds on the state uncertainty set and is followed closely by the sensitivity matrix method. However, even with various enhancements to the formulation of the linear programming problem the computational complexity of this method can be prohibitive for a large class of applications. By contrast, the *Ellipsoidal Bounding* method is very efficient but it is somewhat disappointing in that it produces a rather conservative bounds and it is sensitive to the order of processing of the constraints. We conclude therefore that the ellipsoid technique is best suited for a rapid, rough approximation of the state uncertainty set, while the other two techniques are preferable for calculation of tight and consistent bounds. The proposed *Sensitivity Matrix* method offers a good compromise between the accuracy and efficiency of estimation of the state uncertainty set. It demonstrates computational efficiency comparable to ellipsoidal bounding (and much better than linear programming) and it offers accuracy that is comparable to linear programming results (and much better than ellipsoidal bounding).

REFERENCES

- Arruda, L., Favier, G., Amaral, W. (1991), Proc. of the 1st European Control Conference, France, 1194-1199.
- Bargiela, A. (1985), An algorithm for observability determination in water systems state estimation, *Proc. IEE*, Vol. 132, Pt. D, 6, 245-249.
- Bargiela, A., Hainsworth, G.D. (1989), Pressure and flow uncertainty in water systems, *ASCE J. Water Res. Planning and Management*, 115, 2, 212-229.
- Bargiela, A. (1994), Ellipsoid method for quantifying the uncertainty in water system state estimation, *Proc. IEE Colloquium on Modelling Uncertain Systems*, Vol. 1994/105, 10/1-10/3.
- Bargiela, A., (2001) Interval and ellipsoidal uncertainty in water system state estimation, in: Granular Computing, (Pedrycz, W., ed.), Physica Verlag, 23-57.
- Belforte, G., Bona, B., Frediani, S. (1984), Proc. of the IEEE Conference on Decision and Control, Las Vegas, NV, 1554-1559.

- Belforte, G., Bona, B. (1985), An improved parameter identification algorithm for signals with unknown-but-bounded errors, *Proc. IFAC/IFORS*, York.
- Cichocki, A., Bargiela, A. (1997), Neural networks for solving linear inequality systems, *Parallel Computing*, Vol. 22, No. 11, 1455-1475.
- Claramunt, C., Jiang, B., Bargiela, A., (2000), A new framework for the integration, analysis and visualization of urban traffic data within geographic information systems, *Transportation Research – Part C*, 167-184.
- Dubois, D., Kerre, E., Mesiar, R., Prade, H. (2000), Fuzzy interval analysis, in Dubois, D., Prade, H., (eds), *The Handbook of Fuzzy Sets*, Kluwer, 483-581.
- Fogel, E., Huang, Y.F. (1982), On the value of information system identification – bounded noise case, *Automatica*, Vol. 18, No. 2.
- Gabrys, B., Bargiela, A. (1999), Neural networks based decision support in presence of uncertainties, *ASCE J. of Water Res. Planning and Management*, Vol. 125, 5, 272-280.
- Hainsworth, G.D (1988), Measurement uncertainty in water distribution telemetry systems, *PhD thesis, The Nottingham Trent University*, UK..
- Hartley, J.K., Bargiela, A. (1997), Parallel state estimation with confidence limit analysis, *Parallel Algorithms and Applications*, Vol., 11, No. 1-2, 155-167.
- Jaulin, L., Walter, E. (1996), Guaranteed nonlinear set estimation via interval analysis, in: *Bounding Approaches to System Identification* (Milanese et al., eds.), Plenum Press.
- Kreinovich, V., Lakeyev, A., Rohn, J., Kahl, P. (1998), *Computational Complexity and Feasibility of Data Processing and Interval Computations*, Kluwer, Dordrecht.
- Kurzhanski, A. B., and Valyi, I. (1997), *Ellipsoidal Calculus for Estimation and Control*, Birkhauser.
- Milanese, M, et al. (eds.) (1996), *Bounding Approaches to System Identification*, Plenum Press, NY.
- Mo, S.H., Norton, J.P. (1988), Parameter bounding identification algorithms for bounded-noise records, *Proc. IEE*, Vol. 135, Pt D, No. 2.
- Moore, R.E. (1966), *Interval analysis*, Prentice-Hall, Englewood Cliffs, NJ.
- Nguyen, H.T., Kreinovich, V., Zuo, Q. (1997), Interval valued degrees of belief: applications of interval computations to expert systems and intelligent control, *Int. J. Uncertainty, Fuzziness and Knowledge-Based Systems*, 5, 317-358.
- Norton, J.P. (1986), *An Introduction to Identification*, Academic Press.
- Pedrycz, W., Gomide, F. (1998), *An introduction to Fuzzy Sets. Analysis and Design*, MIT Press, Cambridge, MA.
- Pedrycz, W., Smith M.H., Bargiela, A. (2000), A Granular Signature of Data, *Proc. NAFIPS'2000*.
- Ratschek, A., Rokne, J. (1988), *New computer methods for global optimization*, Ellis Horwood Ltd., John Wiley & Sons, New York.

Rokne, J.G. (2001), *Interval arithmetics and interval analysis: An introduction*, in: *Granular Computing* (Pedrycz ed.), Physica Verlag.

Ross, T., Kreinovich, V., Joslyn, C. (2000), Assessing the predictive accuracy of complex simulation models, *Proc. IFSA/NAFIPS'2001*, Vancouver, 2008-2012.

Bargiela, A., Pedrycz, W., Tanaka, M. (2002), A Study of Uncertain State Estimation, *IEEE Trans. on Syst. Man and Cybernetics*, to appear.

Schweppe, F.C. (1973), *Uncertain dynamic systems*, Prentice-Hall, Englewood Cliffs, NJ.

Warmus, M. (1956), Calculus of approximations, *Bulletin de l'Academie Polonaise des Sciences*, 9(4), 253-259.

EPILOGUE

There is an interesting general tide of scientific development from detailed empirical information to abstract knowledge building. This is an inherently human way of coping with complexity.

While, undoubtedly, 20th century will remain in our memories as a century of digital technology, it has left us with a realization that the real world does not lend itself to a manageable description/modelling using numerical data only. The vast quantities of raw data that are now available need to be processed, through intelligent abstraction, into higher-level entities, i.e. information granules. This is at the root of the emergence of Granular Computing paradigm. The challenge of Granular Computing is to develop *human-centric* computer technologies that will overcome the limitations of the current *machine-centric* approaches.

This book will have fulfilled its purpose if it contributed to stimulating further research in this exciting and important area that will undoubtedly underpin much of scientific progress in the 21st century.

INDEX

A

- absorption, 24
- A-equality, 90
- analog world, 4
- associativity, 23, 59
- attribute space, 92

B

- balance of uncertainty, 109
- belongingness relationship, 22
- Boston housing data, 151, 226, 248, 277
- bottom A-equality, 90

C

- cardinality, 23, 62
- Cartesian product, 25
- centered enclosures, 41
- characteristic function, 22
- calibration of fuzzy sets, 75
- classes of membership functions, 54
- coding mechanisms, 261
- collaborative clustering, 267, 273
- collaboration space, 295
- compatibility measure, 131
- complement, 22
- communication
 - between granular worlds, 255
 - with a numeric world, 261
- comutativity, 24, 59
- concavity, 53
- confidence limits, 419
- convexity, 53
- convolution of fuzzy sets, 73
- core of fuzzy set, 52

D

- data compression, 3
- data confidentiality, 295
- data security in collaborative clustering, 295
- data-oriented, 4
- decoding, 12, 261
- defuzzification schemes, 263
- degree of preference, 50
- degree of similarity, 50
- degree of uncertainty, 51
- dependency effect, 27
- dimension of information granules, 17
- directional communication, 301
- distributivity, 23

E

- ECG data, 369, 391
- ellipsoid method, 427
- embedding principle, 76
- empty set, 22
- encoding, 12
- enclosure of functions, 40
- entropy measure, 63
- Euclidean distance, 119, 132, 224, 271, 274, 308, 352
- experimental relevance of granules, 119

F

- family of sets, 22
- finite state machine, 329
- formal models, 5
- functional mapping of sets, 25

fuzzy arithmetic, 69
 fuzzy C-means (FCM), 219, 271
 fuzzy JK flip-flop, 330
 fuzzy Moore state machine, 337
 learning scheme, 337
 fuzzy numbers, 69
 fuzzy relations, 71, 401
 fuzzy relational equations, 399
 fuzzy sets, 47
 design, 241
 of type-2 and higher, 101
 of level-2 and higher, 103
 reconstruction, 258
 validation, 244
 fuzzy sets transformations, 67
 fuzzy similarity relation, 105, 194
 fuzzy state machine, 328

G

Gaussian membership function, 111
 geometry of fuzzy sets, 51
 GIS, 2
 glass data, 363
 granular analysis, 141
 granules
 compatibility, 131
 inclusion, 139
 relevance, 6
 usefulness, 7
 granular clustering, 128
 interpretation, 130
 validation, 130
 granular computing, 9
 granular data compression, 399
 granular models of signals, 387
 predictive description, 388
 condensation of numeric
 signals, 388
 granular prototyping, 193
 granular time series, 179
 granular worlds, 8
 communication, 11, 255
 networking, 267
 granulation of spatial structures, 163

H

Hamming distance, 119, 132, 241,
 352, 403
 hesitancy, 116
 horizontal collaboration, 269
 horizontal collaborative clustering,
 270
 human-centric, 4
 hyperbox, 34
 characterization, 142
 hyperbox power set, 28

I

identification set, 23
 idempotence, 24
 image compression, 402
 inclusion measure, 139
 information granulation, 126
 information granules, 126
 assessment, 174
 characterization, 164
 design, 164, 237
 interpretation, 174
 reconstruction, 257
 stability, 235
 validation, 237
 information processing pyramid, 9
 information systems, 84
 intelligent agents, 323
 model of, 328
 interoperability, 15
 intersection, 23
 interval
 analysis, 29
 center, 30
 matrices, 36
 operations, 30
 state estimation, 417
 vectors, 34
 width, 30
 interval-valued fuzzy sets, 99, 122
 intuitionistic fuzzy sets, 115
 inverse similarity problem, 210
 involution, 23

J

JAVA mascot, 407

K

knowledge-oriented, 4

L

Lagrange multipliers, 219, 273, 287, 306

law of contradiction, 24

law of excluded middle, 23

linear programming method, 422

linguistic variables, 66

logic-based fuzzy clustering, 217

logic processor, 335

logic transformation of information granules, 304

Lukasiewicz implication, 221

M

membership function, 14, 49, 51, 75, 101, 130, 224, 237, 263, 304, 327, 380, 400

classes of (Gaussian, parabolic, triangular), 54

measurement uncertainty set, 420

monotonicity, 59

Monte Carlo method, 420

Moore type fuzzy state machines, 334

multivalued logic, 74

multivalued implication, 220

N

necessity measure, 65

networking of granular worlds, 267

O

operations on fuzzy sets, 58

overlap index, 140

optimization of similarity levels, 209

P

parabolic membership function, 110

performance index, 196

phase-space granulation, 183

power set, 23, 28, 34

possibility measure, 65

principle of granular clustering, 128

probabilistic sets, 114

probability of granular constructs, 119

projection operation, 25

prototype optimization, 198

pseudocomplement, 400

Q

quantification of collaborative phenomenon, 276

R

reciprocal image of a set, 26

recursive information granulation, 161

reducts, 92

relational calculus, 71, 402

relative complementation, 24

residuation, 400

rough functions, 93

rough sets, 81

approximations, 87

characterization, 88

comparisons, 90

signal granulation, 395

S

s-norm, 59, 117, 194, 210, 257, 309, 400

self-organizing maps (SOM), 349

cluster growing, 354

data distribution map, 357

region map, 356

weight map, 355

sensitivity matrix method, 433

set approximation of fuzzy sets, 239, 256

set-based granules, 164
 set enclosure, 26
 set identifier, 23
 set operations, 23
 set theory, 19
 shadowed fuzzy sets, 122
 shadowed sets, 107
 operations, 112
 transformations, 113
 signal analysis, 377
 similarity of fuzzy sets, 194, 209
 space subdivision enclosures, 42
 sparsity index, 143
 spatial granulation, 2, 378
 specificity, 64
 state uncertainty set, 419
 subtraction operation, 24
 support of fuzzy set, 51
 systems modelling, 417
 swapping techniques, 3

T

t-norm, 59, 117, 194, 210, 257, 309, 400
 temporal granulation, 2, 377
 time-domain granulation, 179
 top A-equality, 90
 triangular membership function, 108
 truncation function, 102
 Tschebyshev distance, 119, 352

U

uncertainty interval, 419
 uncertain system equation, 419
 unimodal, 53
 union, 23
 universal set, 22
 universe of discourse, 22, 75

V

vertical collaboration, 268
 vertical collaborative clustering, 284

W

water distribution system, 436
 wine data, 361
 wrapping effect, 28

$\mathcal{P}(A)$ power set of A , 23, 28, 34
 α -cut, 52, 162, 240
 $\mu_A(x)$ or $A(x)$ membership function, 49, 51
 σ -count, 63, 252, 326
 $\chi_A(x)$ characteristic function, 22, 58, 76, 166, 235
 χ^2 statistic, 252